AD-A062 916 CATHOLIC UNIV OF AMERICA WASHINGTON D C DEPT OF PHYSICS F/G 9/2
A FORTRAN CODE FOR THE CALCULATION OF SOUND PROPAGATION IN A RA--ETC(U)
JAN 78 F S CHWIEROTH, G L ZARUR, A NAGAL N00173-77-C-0008

UNCLASSIFIED NL

1 OF 1

AD
A062916

END
DATE
FILMED
3 -- 79
DDC

LEVEL III

AD062915

ADA062916

DDC FILE COPY

DEPARTMENT OF PHYSICS

The Catholic University of America

Washington, D.C. 20064

78 12 14 040

LEVEL III

(1)

(6) A FORTRAN Code for the Calculation of Sound Propagation in a Range Dependent Ocean, Volume II, Range Functions and Mode Coupling.

(First Version)

(12) 57 p.

F.S.Chwieroth* and A. Nagl
Department of Physics
Catholic University of America
Wahington DC 20064

(10) F.S./Chwieroth, G.L./Zarur, Anton/Nagl H./Ueberall

Versar Inc.
Springfield VA 22151

(15) N00173-77-C-0008

H. Überall
Department of Physics
Catholic University of America
Wahington DC 20064

(11) 14 Jun 78

(16) F52552 (17) SF52552691

Supported by Code 8120, Naval Research Laboratory, Washington DV 20375 under Contract No. N00~~013~~173-77-C-0008

(9) Final rept. 15 Oct 76-15 Jan 77,

* Now at Singer Co., Link Division, Silver Spring MD

076 433

78 12 14 040

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A FORTRAN CODE FOR THE CALCULATION OF SOUND PROPAGATION IN A RANGE DEPENDENT OCEAN II. RANGE FUNCTIONS AND MODE COUPLING.	5. TYPE OF REPORT & PERIOD COVERED Technical Report (final) Oct.15,1976-Jan.15,1977
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) F.S.Chwieroth, A.Nagl, G.L.Zarur, and H. Uberall	8. CONTRACT OR GRANT NUMBER(s) NRL N00173-77-C-0008
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Physics Catholic University of America Washington DC 20064	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program 62759N, Project SF52-552, Task SF52-552-691, Work Unit S01-94.802
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Sea Systems Command Washington DC 20375	12. REPORT DATE June 14, 1978
	13. NUMBER OF PAGES iii+22 pp & pp program listing
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Code 8121 Naval Research Laboratory WAshington DC 20375	15. SECURITY CLASS. (of this report) unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Sound Propagation; Ocean Acoustics; Range dependent Ocean; Range Functions; Arbitrary depth and range dependence; FORTRAN code; Normal modes; Near-Separation; Mode Coupling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Pierce's adiabatic normal-mode theory of sound propagation in a range dependent ocean duct is extended to the case of sound channels with less than gradual range dependence. A computational method is devised to solve the ensuing coupled range equations by diagonalization. This is applied to a channel with arbitrary (numerically given) range dependence, by dividing it into range segments with constant properties each. The local depth functions are taken as the Airy function solutions in a piece-wise linear-

ized stratified medium with arbitrary sound velocity profile. The method is applied to some simple as well as to realistic cases of range dependent sound channels, and the dependence of the mode coupling effects on the degree of range dependence is determined.

Appended to the theory part exhibited in this present first version of the report is a listing of a computer program based on the foregoing theory; it computes range functions and solves the coupled-mode program of sound propagation in a range dependent ocean.

Mode Coupling in a Range Dependent

Under-Ocean Sound Channel

G. L. Zarur

Versar Inc., Springfield VA 22151

A. Nagl and H. Überall

Department of Physics, Catholic University, Washington, D.C. 20064*

F. S. Chwieroth

Singer Company, Link Division, Silver Spring, Md. 20904

Abstract

Pierce's adiabatic normal-mode theory of sound propagation in a range dependent ocean duct is extended to the case of a sound channel with less than gradual range dependence. A computational method is devised to solve the ensuing coupled range equations by diagonalization. This is applied to a channel with arbitrary (numerically given) range dependence, by dividing it into range segments with constant properties each. The local depth functions are taken as the Airy function solutions in a piece-wise linearized stratified medium with arbitrary sound velocity profile. The method is applied to some simple as well as to realistic cases of range dependent sound channels, and the dependence of the mode coupling effects on the degree of range dependence is determined.

* Supported by Code 8120, Naval Research Laboratory, Washington,D.C. 20375

Introduction

Normal mode theory of underwater sound propagation can, in its original form[1], be applied only to situations where both the ocean floor and the sound velocity profile do not depend on the horizontal coordinate (range coordinate), so that the wave equation may be separated into modal range functions and depth functions. In a realistic ocean, this is very rarely the case, but an approximate method has been developed by Pierce[2] in which gradually range dependent environments can be handled to first approximation. This "adiabatic" approximation, based on the Born-Oppenheimer method of molecular physics, utilizes a modal expansion in which the depth functions are taken as the "local" modes of a locally stratified ocean which depend parametrically on range. Insertion into the wave equation then leads to a set of coupled equations for the range functions, but these become uncoupled if in the adiabatic approximation, the coupling terms are neglected. The adiabatic method has been applied in our previous work to a wedge-shaped continental shelf with constant sound velocity[3], to a sound channel with parabolic profile that opens up linearly in range[4], and to sound channels with arbitrary (but gradual) dependence of the ocean floor and of the sound speed profile (which may depend arbitrarily on depth also) on the range[5].

No energy transfer between modes takes place in the adiabatic approximation. If the mode coupling terms are not neglected, the range equations are coupled and energy flows between the modes.

This has been discussed on the basis of coupled power equations by McDaniel[6], and was considered by her in a calculation of statistical mode conversion effects as caused by a rough ocean floor[7]. For the deterministic case, the mode coupling problem has been solved in our earlier work[8] on the sound channel with a range dependent parabolic velocity profile, by using diagonalization techniques for the decoupling of the range equations.

In the present work, similar techniques are used for the more general case where the profile depends <u>arbitrarily</u> on both depth and range, and the ocean floor on range. This is achieved by dividing the ocean into horizontal layers as well as range segments. Linearization is employed in the depth layers, and has been applied also to the range dependent quantities in the range segments in our earlier work[5] on the adiabatic case. The present paper, however, employs the approximation of range segment-wise constancy of the depth functions and their eigenvalues as well as of the coupling terms, in order not to encumber the more involved decoupling problem even further. A different approach to the mode coupling problem based on the "slab method"[9], is due to Kanabis[10].

Pierce's method will be implemented for the coupled-mode problem of an arbitrarily range dependent sound channel in the following, and applied to simple as well as to realistic cases in order to determine the dependence of the mode coupling effects on the degree of range dependence.

I. Local Eigenmodes of the Wave Equation

We seek a solution for the time harmonic pressure field $p(\vec{r})$ of a point source, located at the position $\vec{r}_o$ in an inhomogeneous ocean of density $\rho(\vec{r})$ and propagation constant $k(\vec{r}) = \omega / c(\vec{r})$, as determined by the wave equation

$$\rho(\vec{r})\vec{\nabla}\cdot\left[\rho^{-1}(\vec{r})\vec{\nabla}p(\vec{r})\right] + k^2(\vec{r})\,p(\vec{r}) = \delta(\vec{r}-\vec{r}_o). \tag{1a}$$

Adopting a cylindrical coordinate system with the z-axis pointing vertically downward, we shall assume the sound speed $c(\vec{r})$ to depend arbitrarily on both the depth z, and on the horizontal range coordinate $\vec{\rho} \equiv (x, y)$. If the medium (which includes the ocean bottom) is layered, the boundaries of the ℓth layer $z = z^{\pm}_{\ell}(\vec{\rho})$ may also depend on $\vec{\rho}$ while the ocean surface $z = z_s$ is flat. Densities $\rho(z)$ are taken constant within each layer; they were chosen equal to ρ_w in each water layer but could be set equal to a different constant in each bottom layer. Defining a velocity potential $\phi(\vec{r})$ such that

$$\vec{v}(\vec{r}) = -\vec{\nabla}\phi(\vec{r}),$$
$$p(\vec{r}) = \rho(\vec{r})\,\partial\phi(\vec{r})/\partial t \tag{1b}$$

($\vec{v}$ being the particle velocity), one has in each layer a Helmholtz equation

$$\nabla^2\phi(\vec{r}) + k^2(z,\vec{\rho})\,\phi(\vec{r}) = \delta(\vec{r}-\vec{r}_o). \tag{1c}$$

Following Pierce[2], we attempt a solution in the form of a quasi-separated normal-mode sum

$$\phi(\vec{r}) = \sum_n \psi_n(\vec{\rho})\, u_n(z,\vec{\rho}), \tag{2}$$

in which the "local depth functions" $u_n(z,\vec{\rho})$ satisfy at each range point $\vec{\rho}$ (considered as a parameter) the depth equations pertaining to a stratified medium,

$$\frac{\partial^2 u_n(z,\vec{\rho})}{\partial z^2} + K_n^2(z,\vec{\rho})\, u_n(z,\vec{\rho}) = 0. \tag{3}$$

The eigenvalues $k_n(\vec{\rho})$ of the local modes enter through the local vertical wave number

$$K_n(z,\vec{\rho}) = \left[k^2(z,\vec{\rho}) - k_n^2(\vec{\rho})\right]^{1/2} \tag{4}$$

of the nth mode. They are determined by the boundary conditions satisfied by the local depth functions as used in Eq. (2),

$$\begin{aligned} \rho_\ell\, u_n(z_\ell^+,\vec{\rho}) &= \rho_{\ell+1}\, u_n(z_{\ell+1}^-,\vec{\rho}), \\ \left[\partial u_n/\partial z\right]_{z_\ell^+} &= \left[\partial u_n/\partial z\right]_{z_{\ell+1}^-}, \end{aligned} \tag{5}$$

at the boundary between layers ℓ and $\ell+1$ ($z_\ell^+(\vec{\rho})$ being the boundary as approached from the ℓth layer, and $z_{\ell+1}^-(\vec{\rho})$ as approached from the ℓ+1st layer), corresponding to continuity of pressure and normal particle velocity. Inserting Eq.(2) into Eq. (1c) and using the orthogonality of the depth functions which follows from Eqs. (5),

$$\int_{z_s}^{\infty}[\rho(z)/\rho_w]\,u_n(z,\vec{\rho})\,u_m(z,\vec{\rho})\,dz = \delta_{nm}\,, \tag{6}$$

we obtain, following our earlier derivation[5], the coupled system of (partial differential) range equations

$$\begin{aligned}[\nabla_\rho^2 + k_n^2(\vec{\rho})]\psi_n(\vec{\rho}) &= \delta(\vec{\rho})[\rho(z_0)/\rho_w]u_n(z_0,0)\\ &-2\sum_m[\vec{\nabla}_\rho\psi_m(\vec{\rho})]\cdot\vec{M}'_{nm}(\vec{\rho}) - \sum_m \psi_m(\vec{\rho})M''_{nm}(\vec{\rho}),\end{aligned} \tag{7}$$

in which the coupling coefficients

$$\vec{M}'_{nm}(\vec{\rho}) = \int_{z_s}^{\infty}[\rho(z)/\rho_w]u_n(z,\vec{\rho})\vec{\nabla}_\rho u_m(z,\vec{\rho})\,dz \tag{8a}$$

$$M''_{nm}(\vec{\rho}) = \int_{z_s}^{\infty}[\rho(z)/\rho_w]u_n(z,\vec{\rho})\nabla_\rho^2 u_m(z,\vec{\rho})\,dz \tag{8b}$$

couple the nth and the mth modes. We here used the notation $\vec{\nabla}_\rho \equiv (\partial/\partial x, \partial/\partial y, 0)$, and took the source position as $\vec{r}_0 = (z_0, \vec{\rho}_0 \equiv 0)$. The derivation of Eq. (7), due to our use of Eqs. (5), holds also for the present case where sloping layer boundaries are admitted. However, Eqs. (5) will guarantee the boundary conditions of the total acoustic field,

$$\begin{aligned}\rho_\ell\,\phi^{(\ell)}(z_\ell^+,\vec{\rho}) &= \rho_{\ell+1}\,\phi^{(\ell+1)}(z_{\ell+1}^-,\vec{\rho}),\\ [\partial\phi^{(\ell)}/\partial n]_{z_\ell^+} &= [\partial\phi^{(\ell+1)}/\partial n]_{z_{\ell+1}^-}\,,\end{aligned} \tag{9}$$

to be approximately satisfied only for gradually sloping boundaries and/or small relative density differences between layers,[11] i.e.,

$$[(\rho_{\ell+1} - \rho_\ell)/\rho_{\ell+1}]\tan\alpha \ll 1 \tag{10}$$

(where α is the angle between the boundary normal and the z-axis), as can easily be shown. This will be assumed in the following. Note, however, that no restrictions are imposed on the range gradient of the velocity profile.

The coupling terms of Eqs. (8) may be recast in a simpler form[3] which shows their origin from the range dependence of the sound velocity profile and of the boundaries, respectively, by employing partial intergration and using Eq. (3), with the following results. For the coupling terms which are small of first order in the range dependence, one has

$$\vec{M}'_{nm}(\vec{\rho}) = \vec{C}_{nm}(\vec{\rho}) + \vec{C}^{(s)}_{nm}(\vec{\rho}) \qquad (m \neq n) \tag{11a}$$

$$\vec{M}'_{nn}(\vec{\rho}) = \vec{C}^{(s)}_{nn}(\vec{\rho}) \tag{11b}$$

with the volume contribution

$$\vec{C}_{nm}(\vec{\rho}) = \vec{M}_{nm}(\vec{\rho}) / [k^2_m(\vec{\rho}) - k^2_n(\vec{\rho})], \qquad (m \neq n) \tag{12a}$$

$$\vec{M}_{nm}(\vec{\rho}) = \int_{z_s}^{\infty} [\rho(z)/\rho_w]\, u_n(z,\vec{\rho}) \left[\vec{\nabla}_\rho k^2(z,\vec{\rho})\right] u_m(z,\vec{\rho})\, dz, \qquad (12b)$$

and the surface contributions

$$\tau_{nm}^{(s)}(\vec{\rho}) = M_{nm}^{(s)}(\vec{\rho}) / [k_m^2(\vec{\rho}) - k_n^2(\vec{\rho})], \qquad (m \neq n) \qquad (13a)$$

$$\vec{M}_{nm}^{(s)}(\vec{\rho}) = \sum_{\ell=1}^{L} (\rho_\ell/\rho_W)\left[u_n\, \partial\vec{U}_m/\partial z - \vec{U}_m\, \partial u_n/\partial z\right]_{z_\ell^-(\vec{\rho})}^{z_\ell^+(\vec{\rho})} \qquad (13b)$$

where L is the number of layers, and

$$\vec{U}_m(z,\vec{\rho}) = \vec{\nabla}_\rho\, u_m(z,\vec{\rho}) \; ; \qquad (13c)$$

further,

$$\tau_{nn}^{(s)}(\vec{\rho}) = -\frac{1}{2}\sum_{\ell=1}^{L} (\rho_\ell/\rho_w)\left[u_n^2(z,\vec{\rho})\, \vec{\nabla}_\rho z\right]_{z_\ell^-(\vec{\rho})}^{z_\ell^+(\vec{\rho})} . \qquad (13d)$$

For the second-order coupling coefficients, we shall only quote the results for the volume contributions:

$$M''_{nm}(\vec{\rho}) = N''_{nm}(\vec{\rho}) + \vec{\nabla}_\rho \cdot \vec{M}'_{nm}(\vec{\rho}), \qquad (14a)$$

$$N''_{nm}(\vec{\rho}) = -\sum_{p \neq n,m} \vec{\tau}_{pn}(\vec{\rho}) \cdot \vec{\tau}_{pm}(\vec{\rho}) . \qquad (14b)$$

The surface contributions are absent if the boundaries are flat.

II. Solution of the Coupled Range Equations

In our previous work on mode propagation in a range dependent environment[3-5,8], two important special cases were considered in which range dependence occurs either in the Cartesian x-direction only (x-case), or in the $\vec{\rho}$-direction of cylindrical coordinates only (ρ -case). For sufficient distances from the source, the results of these two assumptions were shown to coincide[4]. We shall here restrict ourselves to the ρ-case only, so that $\psi_n(\vec{\rho}) \equiv \psi_n(\rho)$, and introduce the new range function $f_n(\rho)$ via

$$\psi_n(\rho) = \rho^{-1/2} f_n(\rho). \tag{15}$$

Inserting in Eq. (7) leads to the coupled system of ordinary differential equations

$$f_n''(\rho) + (2\rho)^{-2} f_n + k_n^2(\rho) f_n = \tag{16a}$$

$$= [\rho(z_0)/\rho_W] (2\pi\rho^{1/2})^{-1} \delta(\rho) u_n(z_0,\rho) - 2\sum_m M'_{nm} f'_m - \sum_m V_{nm} f_m$$

where $M'_{nm}(\rho)$ is defined similarly to $\vec{M}'_{nm}$ in Eqs. (11)-(13) but with $\vec{\nabla}_\rho$ replaced by $d/d\rho$, and where the further coupling terms are

$$V_{nm}(\rho) = M''_{nm}(\rho) - \rho^{-1} M'_{nm}(\rho). \tag{16b}$$

Note that as shown before[4], the term $(2\rho)^{-2} f_n$ in Eq. (16a) is negligible for distances from the source in excess of $\sim$1 km, for frequencies up to several Hz. Similarly, the second term in Eq. (16b) is negligible as compared with the second-last term in Eq. (16a)

in an even wider region. We write Eq. (16a) in matrix notation as

$$[d^2/d\rho^2 + (2\rho)^{-2} + k^2(\rho)]\vec{f}(\rho) = \vec{U} - 2M'\vec{f}' - V\vec{f} \tag{16c}$$

where $\vec{f}(\rho)$ is a column vector with elements $f_n(\rho)$, and $k^2(\rho)$ a diagonal matrix with diagonal elements $k_n^2(\rho)$; $\vec{U}$ is the column vector of the source, with components

$$U_n = [\rho(z_0)/\rho_w](2\pi\rho^{1/2})^{-1}\delta(\rho)\,u_n(z_0, 0). \tag{16d}$$

Our goal will be to devise a transformation which uncouples the system of Eq. (16c), at least in an approximate manner. This will be accomplished by first dividing our range ρ into a finite number of inter als with constant vertical boundaries, across which $\rho(z)\phi$ and $\partial\phi/\partial\rho$ have to be continous. In the approximation of Eq. (10), this is shown by integrating over z and using orthonormality of u_n, to lead to continuity of f_n and $df_n/d\rho$, i.e. to

$$f_n(\rho_i^+) = f_n(\rho_{i+1}^-) \tag{17}$$

$$[df_n(\rho)/d\rho]_{\rho_i^+} = [df_n(\rho)/d\rho]_{\rho_{i+1}^-}$$

where ρ_i^+ is the outer boundary of the ith range interval, coinciding with the inner boundary ρ_{i+1}^- of the $i+1$ st range interval.

In our earlier study[5] of the "adiabatic" solution of Eq. (17) where the coupling terms were neglected, we approximated the quantity $k_n^2(\rho) + 1/4\rho^2$ in each range by the linear expressions $C_n\rho + D$, joined continuously at each segment boundary where the actual values were adopted, except in the first (source) interval and in the last one (reaching to infinity), where the constant values $k_n^2(o)$ and $k_n^2(\infty)$, respectively, were adopted for $k_n^2(\rho)$. This leads to Airy function solutions in the general intervals, and to Hankel functions $H_o^{(1)}$ in the two extreme intervals, which become trigonometric solutions if the term $1/4\rho^2$ is neglected. This procedure would also be possible in the present case; the coupling terms would then have the form, e.g.,

$$\mathcal{C}_{nm}(\rho) = M_{nm} / [(C_m - C_n)\rho + D_m - D_n] , \tag{18}$$

with M_{nm} depending on ρ only through u_n and u_m, since $k^2(z,\rho)$ is also being approximated by linear expressions $C(z)\rho+D(z)$ in order to represent the experimental range (and depth) dependent sound speed profile which is given to us only at a finite number of range points (chosen as our range interval boundaries). We shall, however, for reasons of greater simplicity approximate in each range interval the eigenvalues $k_n^2(\rho)$ and depth functions $u_n(z,\rho)$, and hence also the coupling terms, by their values at the midpoints of each range interval, insuring accuracy by introducing a finer subdivision into range intervals if necessary.

We are then able to uncouple Eq. (16c) by matrix transformations involving constant transformation matrices in each range interval. Following our previous approach[8] used in the treatment of mode coupling for a range dependent parabolic profile, we first set

$$\vec{f}(\rho) = S_1 \vec{\Gamma}(\rho) \tag{19}$$

which, with a constant matrix S_1, leads to

$$[d^2/d\rho^2 + K^2(\rho)]\vec{\Gamma} = S_1^{-1}\vec{U} - 2\lambda\vec{\Gamma}' - W\vec{\Gamma} \tag{20}$$

where we neglected $1/4\rho^2$, and called

$$K^2(\rho) = S_1^{-1} k^2(\rho) S_1 \tag{21a}$$

$$\lambda = S_1^{-1} M' S_1 \tag{21b}$$

$$W = S_1^{-1} V S_1 . \tag{21c}$$

The matrix S_1 will be so chosen that λ is diagonal, with diagonal matrix elements that will be called λ_n . (This is possible with a constant matrix S_1 since as mentioned above, M' is taken constant in each range segment). Next, the derivative terms $\vec{\Gamma}'$ which now multiply the diagonal matrix λ , may be eliminated by standard methods. We make the transformation

$$\vec{\Gamma} = \sigma\vec{\gamma} \tag{22a}$$

where σ is taken to be a diagonal matrix with elements σ_n .

Substituting Eq. (22a) into Eq. (20) and equating the coefficients of γ_n' to zero, we find

$$\sigma_n = e^{-\lambda_n \rho} \quad ; \tag{22b}$$

constant factors in Eq. (22b) may all be chosen equal to unity.

The transformed range equations are still coupled (although without first derivatives), and have the form

$$d^2\vec{\gamma}/d\rho^2 - \lambda^2\vec{\gamma} + (\tilde{K}^2 + \tilde{W})\vec{\gamma} = S_1^{-1}\vec{U}. \tag{23}$$

The source term is unaffected by the transformation of Eq. (22a) since $\vec{U}$ contains a delta function in ρ , and $\sigma(\rho=0)$ is the identity matrix. The elements of the non-diagonal matrices $\tilde{K}^2$ and $\tilde{W}$ are given by

$$\tilde{K}^2_{nm} = e^{(\lambda_n-\lambda_m)\rho} K^2_{nm} , \tag{24a}$$
$$\tilde{W}_{nm} = e^{(\lambda_n-\lambda_m)\rho} W_{nm} .$$

In the spirit of our approximation procedure, these will again be replaced by their values at the midpoints of the ρ-intervals, rendering the expression

$$T = \tilde{K}^2 + \tilde{W} - \lambda^2 \tag{25}$$

a constant matrix. The ensuing equation

$$d^2\vec{\gamma}/d\rho^2 + T\vec{\gamma} = S_1^{-1}\vec{U} \tag{26}$$

can be decoupled with one final transformation:

$$\vec{\gamma} = S_2\vec{P} \tag{27}$$

where S_2 is chosen such that

$$S_2^{-1} T S_2 \equiv \Lambda \tag{28}$$

is a diagonal matrix with elements Λ_n . This has finally brought us to the uncoupled system of linear equations with constant coefficients

$$d^2\vec{P}/d\varrho^2 + \Lambda\vec{P} = \vec{U}^* , \quad (29$$

where the new source term is

$$\vec{U}^* = (S_1 S_2)^{-1} \vec{U} . \tag{29b}$$

Its general solution is in the source-free region:

$$\vec{P} = \sum_{i=1}^{N} \vec{\pi}_i (\alpha_i \cos q_i \varrho + \beta_i \sin q_i \varrho), \tag{30a}$$

where N is the total number of modes,

$$q_i = \Lambda_i^{1/2} , \tag{30b}$$

$\vec{\pi}_i$ are the basis vectors with components

$$\pi_{im} = \delta_{im} , \tag{30c}$$

and α_i, β_i are 2N arbitrary coefficients. If we call the overall transformation matrix

$$U = S_1 \sigma S_2 , \tag{31a}$$

our original range functions of Eq. (15) are obtained as

$$\vec{f} = U\vec{P} , \tag{31b}$$

with components

$$f_n(\rho) = \sum_m U_{nm} (\alpha_m \cos q_m \rho + \beta_m \sin q_m \rho). \tag{31c}$$

The Green's function of Eq. (29a) which describes the behavior of the solution at the source can be obtained from Eq. (31c) in the well-known way,[3-5] see below.

As the last step, we shall have to carry out a matching of the range functions and their derivatives at the range interval boundaries, see Eq. (17). In the last interval which reaches out to infinity, we assume no range dependence so that $k_p(\rho)$=constant $=k_p(\infty)$, and have the solution[5]

$$f_p^{\infty}(\rho) = A_p \rho^{1/2} H_0^{(1)}(k_p(\infty)\rho) \tag{32a}$$

(no mode coupling takes place here), comprising only outgoing waves with undetermined modal amplitudes A_p. In the general (ith) interval, we adopt Eq. (31c) with superscripts i on f_n, U_{nm}, α_m, β_m and q_m: in matrix notation, this gives

$$\vec{f}^{\,i}(\rho) = U^i \vec{\delta}^{\,i}(\rho), \tag{32b}$$

where $\vec{\delta}^{\,i}(\rho)$ is a column vector with components

$$\delta_q^i(\rho) = \alpha_q^i \cos q_q^i \rho + \beta_q^i \sin q_q^i \rho. \tag{32c}$$

Introducing the diagonal matrix A with diagonal elements A_p, and the vector $\vec{H}(\rho)$ with components $H_p(\rho) = \rho^{1/2} H_0^{(1)}(k_p(\infty)\rho)$, Eq. (32a) reads in matrix form

$$\vec{f}^{\,\infty}(\rho) = A\vec{H}(\rho). \tag{32d}$$

If we designate the last range interval before the infinite interval by i=M, the matching conditions are (at the boundary $\rho=\rho_M$):

$$\begin{aligned} U^M \vec{\mathfrak{J}}^M(\rho_M) &= A\vec{H}(\rho_M) \\ U^M \vec{\mathfrak{J}}^{M\prime}(\rho_M) &= A\vec{H}'(\rho_M). \end{aligned} \tag{33}$$

We introduce the two-component vectors (2D space)

$$\vec{\alpha}_q^{\,M} = \begin{pmatrix} \alpha_q^M \\ \beta_q^M \end{pmatrix}, \qquad \vec{H}_p(\rho_M) = \begin{pmatrix} H_p(\rho_M) \\ H_p'(\rho_M) \end{pmatrix} \tag{34a}$$

and the 2 x 2 matrix

$$\mathcal{C}_q^{M\rho_M} = \begin{pmatrix} \cos q_q^M \rho_M & \sin q_q^M \rho_M \\ -q_q^M \sin q_q^M \rho_M & q_q^M \cos q_q^M \rho_M \end{pmatrix} \tag{34b}$$

with inverse

$$(\mathcal{C}_q^{M\rho_M})^{-1} = \begin{pmatrix} \cos q_q^M \rho_M & -\sin q_q^M \rho_M / q_q^M \\ \sin q_q^M \rho_M & \cos q_q^M \rho_M / q_q^M \end{pmatrix}. \tag{34c}$$

Then, Eqs. (33) are solved by

$$\vec{\alpha}_q^{\,M} = \sum_p (U_M^{-1}A)_{qp}\, (\mathcal{C}_q^{M\rho_M})^{-1}\, \vec{H}_p(\rho_M). \tag{35a}$$

The notation may be further compacted by intooducing hypervectors $\overleftrightarrow{\alpha}^M$, $\overleftrightarrow{H}(\rho_M)$ whose components (each corresponding to a mode) are the 2-component vectors $\vec{\alpha}_q^M$, $\vec{H}_q(\rho_M)$, respectively, as well as diagonal hypermatrices $\tilde{\mathcal{C}}$ whose diagonal elements are the 2 x 2 matrices $\mathcal{C}_q$ (so that the elements of the diagonal hypermatrix $\tilde{\mathcal{C}}^{-1}$ are the 2 x 2 matrices $\mathcal{C}_q^{-1}$). Eq. (35a) then reads

$$\overleftrightarrow{\alpha}^M = (\tilde{\mathcal{C}}^{M\rho_M})^{-1} U_M^{-1} A \overleftrightarrow{H}(\rho_M), \tag{35b}$$

where the previous mode-space matrices U_M^{-1} and A are understood as being diagonal in 2D space.

Applying Eq. (17) for the match between intervals i and i+1, we have

$$\begin{aligned} U^i \vec{\mathcal{J}}^i(\rho_i) &= U^{i+1} \vec{\mathcal{J}}^{i+1}(\rho_i) \\ U^i \vec{\mathcal{J}}^{i\,\prime}(\rho_i) &= U^{i+1} \vec{\mathcal{J}}^{i+1\,\prime}(\rho_i) \end{aligned} \tag{36}$$

which, in our hyperspace, is solved by

$$\overleftrightarrow{\alpha}^i = (\tilde{\mathcal{C}}^{i\rho_i})^{-1} U_i^{-1} U^{i+1} \tilde{\mathcal{C}}^{i+1,\rho_i} \overleftrightarrow{\alpha}^{i+1}. \tag{37}$$

Finally, we match at the boundary between invervals i=1 (source interval) and i=2. In the former, there is no coupling, and the solution is[5]

$$f_p(\rho) = A_p^{(0)} \rho^{1/2} \left[\alpha_p^1 H_0^{(1)}(k_p(0)\rho) + \beta_p^1 H_0^{(2)}(k_p(0)\rho) \right] \tag{38}$$

where $A_p^{(o)}$ are ~~known~~ unknown amplitude factors which we again arrange in a diagonal matrix $A^{(o)}$. Introducing

$$H_p^{(1,2)}(\rho) \equiv \rho^{1/2} H_o^{(1,2)}(k_p(o)\rho) \tag{39a}$$

and forming the known matrix

$$\mathcal{H}_p = \begin{pmatrix} H_p^{(1)}(\rho_1) & H_p^{(2)}(\rho_1) \\ H_p^{(1)\prime}(\rho_1) & H_p^{(2)\prime}(\rho_1) \end{pmatrix}, \tag{39b}$$

we obtain from Eqs. (17):

$$A^{(o)} \overleftrightarrow{\alpha}^1 = \tilde{\mathcal{H}}^{-1} U_2 \tilde{\mathcal{C}}^{2,\rho_1} \overleftrightarrow{\alpha}^2, \tag{40}$$

where hypermatrix notation was introduced as before. In this way, all coefficients α_q^i and β_q^i have been determined; they are in the ith interval:

$$\overleftrightarrow{\alpha}^i = (\tilde{\mathcal{C}}^{i,\rho_i})^{-1} U_i^{-1} U^{i+1} \tilde{\mathcal{C}}^{i+1,\rho_i} \cdot (\tilde{\mathcal{C}}^{i+1,\rho_{i+1}})^{-1} U_{i+1}^{-1} U^{i+2}$$
$$\tilde{\mathcal{C}}^{i+2,\rho_{i+1}} \ldots (\tilde{\mathcal{C}}^{M,\rho_M})^{-1} U_M^{-1} A \overleftrightarrow{H}(\rho_M), \tag{41a}$$

and in the source interval:

$$A^{(o)} \overleftrightarrow{\alpha}^1 = \tilde{\mathcal{H}}^{-1} U^2 \tilde{\mathcal{C}}^{2,\rho_1} \cdot (\tilde{\mathcal{C}}^{2,\rho_2})^{-1} U_2^{-1} U^3 \tilde{\mathcal{C}}^{3,\rho_2} \ldots$$
$$\ldots (\tilde{\mathcal{C}}^{M-1,\rho_{M-1}})^{-1} U_{M-1}^{-1} U^M \tilde{\mathcal{C}}^{M,\rho_{M-1}} \cdot (\tilde{\mathcal{C}}^{M,\rho_M})^{-1} U_M^{-1} A \overleftrightarrow{H}(\rho_M).$$

(41b)

These expressions still contain the unknown coefficients $A^{(o)}$ and A,

but as was shown earlier,[5] matching to the source leads to

$$A_p^{(0)} = \frac{u_p(z_0,0)}{4i(\alpha_p^2 - \beta_p^2)} \; \frac{\rho(z_0)}{\rho_w} , \tag{42}$$

so that in principle, the remaining unknowns A_p (the outgoing mode amplitudes at infinity) can be determined by the requirement that Eqs. (41b) and (42) be satisfied. The implementation of this prescription, however, requires an iterative approach which will be described below.

III. Numerical Results and Discussion

The above procedure will now be applied to the calculation of transmission loss in some examples of channels with simple and also with realistic velocity profiles. In order to be able to handle the latter case, we shall use the previously employed method[5] for calculating local depth functions $u_n(z,\rho)$ and their eigenvalues $k_n^2(\rho)$ in a numerically given profile, which consists in dividing the ocean into (horizontally bounded) depth segments or layers, in each of which $k^2(z,\rho)$ is approximated linearly, so that the depth functions are given by Airy functions. The layer below the (locally flat) ocean floor is simply modeled as a liquid with constant bottom density ρ_B, and with a z-independent local sound velocity $c_B(\rho)$, which reaches down to $z \to \infty$. The eigenvalues $k_n^2(\rho)$ are then obtained by satisfying the boundary conditions of a pressure release surface at the ocean surface $z=z_s$, i.e.,

$u_p(0,\vec{\rho})$ =0, and of an exponetially decaying solution in the bottom layer. Further details are given in Reference 5.

REFERENCES

1. See, e.g., I.S. Tolstoy and C.S. Clay, Ocean Acoustics (McGraw-Hill, New York, 1966)
2. A.D. Pierce, J. Acoust. Soc. Am. 37, 19(1965)
3. R.D. Graves, A. Nagl, H. Überall, and G.L. Zarur, J. Acoust. Soc. Am. 58, 1171 (1975)
4. R. D. Graves, A. Nagl, H. Überall, and G.L. Zarur, Acustica (in press)
5. A. Nagl, H. Überall, A.J. Haug and G.L. Zarur, J. Acoust. Soc. Am. (in press)
6. S.T. McDaniel, J. Acoust. Soc. Am. 60, 1285 (1976)
7. S.T. McDaniel, J. Acoust. Soc. Am. 62, 320 (1977), and to be publ.
8. F. S. Chwieroth, R. D. Graves, A. Nagl, H. Überall, and G.L. Zarur, submitted to J. Acoust. Soc. Am.
9. J.R. Wait and K.P. Spies, Radio Sci. 3, 787 (1968)
10. W.G. Kanabis, Naval Underwater Systems Center Techn. Reports 4319 (1972); 4887-I (1975), 4887-II (1976)

The following computer program is based on the foregoing theory; it computes range functions and solves the coupled-mode program of sound propagation in a range dependent ocean.

The main program GPC (pp. 24 - 32) computes the propagation loss as a function of range and depth. It calls all the subroutines; it calculates the range functions, it calculates the coefficients of the linear combinations of the range functions in all the range segments.

The subroutine NAGL (p.33) interfaces the range function and depth function programs; it calls the depth function program which is described in Volume I of this report.

The subroutine FORM (pp.34-37) calculates the U's and the q-eigenvalues

The subroutine COEF1 (pp.38-40) calculates the mode coefficients in the source segment and in the infinite segment by matching directly (since all the internal boundary conditions are here lumped together into one matrix)

The subroutine HMDIAG (p.41) diagonalizes an antihermitean matrix (by relating it to the diagonalization of a Hermitean one)

The subroutine MXEL (p.42) computes integrals $\int u_n u_m z \, dz$ containing Airy functions

The subroutine HANK (p.43) calls the Hankel function subroutine

The subroutine HANGL (p.43) computes the Hankel functions of argument X

The subroutine DAIRY (p.44-50) is the same as in the depth function program (Vol. I of this report)

The subroutine PLT (p.51-52) is a plot routine to plot the propagation loss

The last page (p.53) tells what has to be loaded into the range, depth, and plotting programs.

```
TY GPC
      COMMON /DEPTH/ XKN(15,4),
     1UO(15)
      COMMON /FORMC/ X([illegible]),Z(4),ZB(4),NM,NL,NLM,NLP,XK(4,4),DEN(4)
     1,XNAG(4)
      DIMENSION XKT([illegible]),[illegible](15,4),
     1U(15,15),UI(15,15),Q(15),AM(30,30),V(15),DUM(30,30),
     2A(15),AO(15),ALF(3,15),BET(3,15),US(2,15,15),UIS(2,15,15),
     3QS(2,15),FR(15),UN(15),IOUT(15),G(30),XKNT(15),NMODE(4)
      COMPLEX U,UI,Q,AM,SUM,EYE,V,DUM,A,AO,ALF,BET,US,UIS,QS,SUM2,
     1FR,UN,C1,S1,H1,H1P,H2,H2P,A1,A2,A3,A4,A5,TEM,QR
      DIMENSION UOD(15),URD(15,4),XKND(15,4)
      DOUBLE PRECISION XNAG,UOD,URD,XKND
      EYE=(0.,1.)
      PI=3.14159
      TYPE 720
  720 FORMAT(' TYPE 0 FOR NO COUPLING, 1 FOR COUPLING')
      ACCEPT 2,ICOUP
      TYPE 1
    1 FORMAT('  INPUT NUMBER OF MODES, LAYERS, AND SEGMENTS')
      ACCEPT 2,NM,NL,NS
      TYPE 2,NM,NL,NS
    2 FORMAT(5G)
      NLM=NL-1
      NLP=NL+1
      NSM=NS-1
      NSP=NS+1
      TYPE 460
  460 FORMAT(' INPUT NUMBER OF MODES FOR EACH PROFILE')
      ACCEPT 2,(NMODE(I),I=1,NSP)
      TYPE 2,(NMODE(I),I=1,NSP)
      TYPE 6
    6 FORMAT('  INPUT FREQUENCY')
      ACCEPT 2,F
      TYPE 2,F
      F=2.*PI*F
      TYPE 700
  700 FORMAT('  INPUT CONVERGENCE CRITERION')
      ACCEPT 2,FRAC
      TYPE 2,FRAC
      TYPE 3
    3 FORMAT('  INPUT SEGMENT POSITIONS')
      X(1)=2000.
      X(2)=20000.
      X(3)=40000.
      TYPE 2,(X(I),I=1,NS)
      ACCEPT 2,(X(I),I=1,NS)
      XNAG(1)=0.
      DO 200 I=1,NS
  200 XNAG(I+1)=X(I)
      TYPE 4
    4 FORMAT('  INPUT SOURCE DEPTH, RECEIVER DEPTH, NL-1 PROFILE',
     1' DEPTHS, NS+1 BOTTOM DEPTHS')
      ACCEPT 2,ZS,ZR,(Z(I),I=2,NL),(ZB(I),I=1,NSP)
      TYPE 2,ZS,ZR,(Z(I),I=2,NL),(ZB(I),I=1,NSP)
      Z(1)=0.
      DO 101 IZS=1,NL
      IF(ZS.LT.Z(IZS+1)) GO TO 8
  101 CONTINUE
    8 TYPE 5
    5 FORMAT('  INPUT SOUND SPEED PROFILE FOR EACH SEGMENT')
      DO 802 IS=1,NSP
  802 XK(1,IS)=1520.
      XK(2,1)=1515.
      XK(2,2)=1514.5
      XK(2,3)=1510.
```

```
      XK(2,4)=1505.
      XK(3,1)=1490.
      XK(3,2)=1490.
      XK(3,3)=1490.
      XK(3,4)=1490.
      XK(4,1)=1540.
      XK(4,2)=1540.
      XK(4,3)=1540.
      XK(4,4)=1540.
      DO 100 IS=1,NSP
C      ACCEPT 2,(XK(IL,IS),IL=1,NLP)
      TYPE 2,(XK(IL,IS),IL=1,NLP)
      DO 100 IL=1,NLP
      XK(IL,IS)=(F/XK(IL,IS))**2
  100 CONTINUE
      TYPE 7
    7 FORMAT('  INPUT RELATIVE DENSITIES FOR EACH LAYER')
      DO 803 IL=1,NLP
  803 DEN(IL)=1.
      TYPE 2,(DEN(IL),IL=1,NLP)
C      ACCEPT 2,(DEN(IL),IL=1,NLP)
      TYPE 40
   40 FORMAT('  TYPE NUMBER OF MODES TO BE OUTPUT')
      ACCEPT 2,NOUT
      TYPE 2,NOUT
      NOUT2=2*NOUT
      IF(NOUT.EQ.0) GO TO 41
      IF(NOUT.EQ.NM) GO TO 43
      TYPE 42
   42 FORMAT('  TYPE WHICH MODES ARE TO BE OUTPUT')
      ACCEPT 2,(IOUT(I),I=1,NOUT)
      GO TO 41
   43 DO 44 I=1,NM
   44 IOUT(I)=I
C
C     DETERMINE DEPTH FUNCTIONS
C
   41 CALL NAGL(NM,NLP,NSP,NMODE,Z,ZB,XK,XKN,UO,UR,ZS,XKND,UOD,URD)
C
C     FORM PRODUCT OF MATRICES ON PAGE 44...U2*H2(X1)*H2I(X2)*U2I*U3*...
C     FOR USE IN CONNECTING SOURCE INTERVAL WITH INFINITE INTERVAL.
C     SIZE OF MATRIX IS NOW TWICE THE NUMBER OF MODES TO ALLOW FOR
C     INGOING AND OUTGOING SOLUTIONS.   MATRIX U IS DIAGONAL IN THE 2
C     SOLUTIONS.   MATRIX SCRIPT-C IS DIAGONAL IN MODE NUMBER.
C
      NM2=2*NM
      DO 106 I=2,NM2
      JM=I-1
      DO 107 J=1,JM
      AM(I,J)=0.
  107 AM(J,I)=0.
  106 AM(I,I)=1.
      AM(1,1)=1.
C      TYPE 201,((AM(I,J),J=1,NM2),I=1,NM2)
  201 FORMAT(' 201 AM'/(12G))
      DO 108 IS=2,NS
      ISM=IS-1
      ISTEM=IS
      CALL FORM(ISTEM,U,UI,Q,ICOUP,NMODE)
C      TYPE 621,(G(I),I=1,NM)
  621 FORMAT (' 621 Q'/6G)
      DO 118 I=1,NM
      DO 118 J=1,NM
      US(ISM,I,J)=U(I,J)
  118 UIS(ISM,I,J)=UI(I,J)
      DO 119 I=1,NM
```

```
  119 QS(ISM,I)=Q(I)
C
C     MULTIPLY FROM RIGHT BY U
C
      DO 109 I=1,NM2
      DO 109 K=1,NM
      DO 109 J=1,2
      SUM=0.
      DO 110 L=1,NM
  110 SUM=SUM+AM(I,2*(L-1)+J)*U(L,K)
  109 DUM(I,2*(K-1)+J)=SUM
C      TYPE 222,((DUM(I,J),J=1,NM2),I=1,NM2)
  222 FORMAT(' 222 DUM'/(12G))
C
C     MULTIPLY FROM RIGHT BY SCRIPT-C EVALUATED AT LEFT OF SEGMENT
C
      DO 111 L=1,NM2
      DO 111 I=1,NM
      I1=2*(I-1)
      C1=CCOS(Q(I)*X(ISM))
      S1=CSIN(Q(I)*X(ISM))
      AM(L,I1+1)=DUM(L,I1+1)*C1-DUM(L,I1+2)*Q(I)*S1
  111 AM(L,I1+2)=DUM(L,I1+1)*S1+DUM(L,I1+2)*Q(I)*C1
C      TYPE 223,((AM(I,J),J=1,NM2),I=1,NM2)
  223 FORMAT(' 223 AM'/(12G))
C
C     MULTIPLY FROM RIGHT BY SCRIPT-C INVERSE EVALUATED AT RIGHT OF
C     SEGMENT
C
      DO 112 L=1,NM2
      DO 112 I=1,NM
      I1=2*(I-1)
      C1=CCOS(Q(I)*X(IS))
      S1=CSIN(Q(I)*X(IS))
      DUM(L,I1+1)=AM(L,I1+1)*C1+AM(L,I1+2)*S1
  112 DUM(L,I1+2)=(-AM(L,I1+1)*S1+AM(L,I1+2)*C1)/Q(I)
C      TYPE 224,((DUM(I,J),J=1,NM2),I=1,NM2)
  224 FORMAT(' 224 DUM'/(12G))
C
C     MULTIPLY FROM RIGHT BY U INVERSE
C
      DO 113 I=1,NM2
      DO 113 K=1,NM
      DO 113 J=1,2
      SUM=0.
      DO 114 L=1,NM
  114 SUM=SUM+DUM(I,2*(L-1)+J)*UI(L,K)
  113 AM(I,2*(K-1)+J)=SUM
C      TYPE 225,((AM(I,J),J=1,NM2),I=1,NM2)
  225 FORMAT(' 225 AM'/(12G))
  108 CONTINUE
C
C     MULTIPLY FROM LEFT BY SCRIPT-H INVERSE
C
      X2=SQRT(X(1))
      DO 115 I=1,NM
      I1=2*(I-1)
      RHO=SQRT(XKN(I,1))*X(1)
      CALL HANK(RHO,H1,H1P,H2,H2P)
C      TYPE 243,I,RHO,H1,H1P,H2,H2P
      XKTEM=RHO/X(1)
      A1=X2*H1
      A2=X2*H2
      A3=XKTEM*X2*H1P+H1/(2.*X2)
      A4=XKTEM*X2*H2P+H2/(2.*X2)
      A5=A1*A4-A2*A3
```

```
      TEM=A1
      A1=A4/A5
      A4=TEM/A5
      A2=-A2/A5
      A3=-A3/A5
      DO 115 L=1,NM2
      DUM(I1+1,L)=A1*AM(I1+1,L)+A2*AM(I1+2,L)
  115 DUM(I1+2,L)=A3*AM(I1+1,L)+A4*AM(I1+2,L)
C       TYPE 226,((DUM(I,J),J=1,NM2),I=1,NM2)
  226 FORMAT(' 226 DUM'/(12G))
C
C     MULTIPLY FROM RIGHT BY H
C
      XNS=SQRT(X(NS))
      DO 116 J=1,NM
      J1=2*(J-1)
      RHO=SQRT(XKN(J,NSP))*X(NS)
      CALL HANK(RHO,H1,H1P,H2,H2P)
C       TYPE 243,J,RHO,H1,H1P,H2,H2P
      XKTEM=RHO/X(NS)
      DO 116 I=1,NM2
  116 AM(I,J)=DUM(I,J1+1)*XNS*H1+DUM(I,J1+2)*(XKTEM*XNS*H1P+H1/(2.*
     1XNS))
C       TYPE 227,((AM(I,J),J=1,NM),I=1,NM2)
  227 FORMAT(' 227 AM'/(6G))
C
C     FORM MATRIX TO BE USED IN DETERMINING COEFFICIENTS IN SOURCE
C     SEGMENT
C
      DO 117 I=1,NM
  117 V(I)=U0(I)*DEN(IZS)/(4.*EYE)
C       TYPE 228,(V(I),I=1,NM)
  228 FORMAT(' 228 V'/(6G))
C
C     FIND COEFFICIENTS FOR SOURCE INTERVAL
C
      CALL COEF1(NM,AM,V,A,A0,ALF,BET,FRAC)
C
C     DETERMINE COEFFICIENTS FOR OTHER INTERVALS USING RELATION ON
C     PAGE 44
C
      DO 120 I=2,NM2
      JM=I-1
      DO 121 J=1,JM
      AM(I,J)=0.
  121 AM(J,I)=0.
  120 AM(I,I)=1.
      AM(1,1)=1.
C       TYPE 240,((AM(I,J),J=1,NM2),I=1,NM2)
  240 FORMAT(' 240 AM'/(12G))
C
C     MULTIPLY FROM RIGHT BY SCRIPT-C INVERSE EVALUATED AT BOUNDARY OF
C     INFINITE SEGMENT
C
      DO 122 L=1,NM2
      DO 122 I=1,NM
      I1=2*(I-1)
      C1=CCOS(QS(NSM,I)*X(NS))
      S1=CSIN(QS(NSM,I)*X(NS))
      DUM(L,I1+1)=AM(L,I1+1)*C1+AM(L,I1+2)*S1
  122 DUM(L,I1+2)=(-AM(L,I1+1)*S1+AM(L,I1+2)*C1)/QS(NSM,I)
C       TYPE 241,((DUM(I,J),J=1,NM2),I=1,NM2)
  241 FORMAT(' 241 DUM'/(12G))
C
C     MULTIPLY FROM RIGHT BY U INVERSE
C
```

```
      DO 123 I=1,NM2
      DO 123 K=1,NM
      DO 123 J=1,2
      SUM=0.
      DO 124 L=1,NM
  124 SUM=SUM+DUM(I,2*(L-1)+J)*UIS(NSM,L,K)
  123 AM(I,2*(K-1)+J)=SUM
C      TYPE 242,((AM(I,J),J=1,NM2),I=1,NM2)
  242 FORMAT(' 242 AM'/(12G))
C
C     MULTIPLY FROM RIGHT BY H EVALUATED AT BOUNDARY OF INFINITE
C     SEGMENT
C
      XNS=SQRT(X(NS))
      DO 125 J=1,NM
      J1=2*(J-1)
      RHO=SQRT(XKN(J,NSP))*X(NS)
      CALL HANK(RHO,H1,H1P,H2,H2P)
C      TYPE 243,J,RHO,H1,H1P,H2,H2P
  243 FORMAT(' 243 J,RHO,H1,H1P,H2,H2P'/10G)
      XKTEM=RHO/X(NS)
      DO 125 I=1,NM2
  125 DUM(I,J)=AM(I,J1+1)*XNS*H1+AM(I,J1+2)*(XKTEM*XNS*H1P+H1/(2.*XNS))
C      TYPE 244,((DUM(I,J),J=1,NM),I=1,NM2)
  244 FORMAT(' 244 DUM'/(6G))
C
C     MULTIPLY FROM RIGHT BY COEFFICIENTS FOR INFINITE SEGMENT TO DEFINE
C     COEFFICIENTS FOR LAST FINITE SEGMENT
C
      DO 126 I=1,NM
      I1=2*(I-1)
      SUM=0.
      SUM2=0.
      DO 127 J=1,NM
      SUM=SUM+DUM(I1+1,J)*A(J)
  127 SUM2=SUM2+DUM(I1+2,J)*A(J)
      AM(I1+1,1)=SUM
      AM(I1+2,1)=SUM2
      ALF(NS,I)=SUM
  126 BET(NS,I)=SUM2
C      TYPE 245,(AM(I,1),I=1,NM2)
  245 FORMAT(' 245 AM'/12G)
C      TYPE 246,(ALF(NS,I),I=1,NM)
  246 FORMAT(' 246 ALF'/6G)
C      TYPE 247,(BET(NS,I),I=1,NM)
  247 FORMAT(' 247 BET'/6G)
C
C     FIND COEFFICIENTS OF OTHER SEGMENTS BY MATCHING SEQUENTIALLY...
C     USE RELATION IN MIDDLE OF PAGE 43
C
      NS2=NS-2
      IF(NS2.LT.1) GO TO 20
      IS=NS
      DO 128 IS2=1,NS2
      IS=IS-1
      ISM=IS-1
C
C     MULTIPLY FROM LEFT BY SCRIPT C FOR PREVIOUS SEGMENT EVALUATED
C     AT BOUNDARY BETWEEN CURRENT AND PREVIOUS SEGMENTS
C
      DO 129 I=1,NM
      I1=2*(I-1)
      C1=CCOS(QS(IS,I)*X(IS))
      S1=CSIN(QS(IS,I)*X(IS))
      DUM(I1+1,1)=C1*AM(I1+1,1)+S1*AM(I1+2,1)
  129 DUM(I1+2,1)=QS(IS,I)*(-S1*AM(I1+1,1)+C1*AM(I1+2,1))
```

```
C        TYPE 248,(DUM(I,1),I=1,NM2)
  248 FORMAT(' 248 DUM'/12G)
C
C       MULTIPLY FROM LEFT BY U FOR PREVIOUS SEGMENT
C
      DO 130 I=1,NM
      I1=2*(I-1)
      DO 130 J=1,2
      SUM=0.
      DO 131 K=1,NM
  131 SUM=SUM+US(IS,I,K)*DUM(2*(K-1)+J,1)
  130 AM(I1+J,1)=SUM
C        TYPE 249,(AM(I,1),I=1,NM2)
  249 FORMAT(' 249 AM'/12G)
C
C       MULTIPLY FROM LEFT BY U INVERSE FOR CURRENT SEGMENT
C
      DO 132 I=1,NM
      I1=2*(I-1)
      DO 132 J=1,2
      SUM=0.
      DO 133 K=1,NM
  133 SUM=SUM+UIS(ISM,I,K)*AM(2*(K-1)+J,1)
  132 DUM(I1+J,1)=SUM
C        TYPE 250,(DUM(I,1),I=1,NM2)
  250 FORMAT(' 250 DUM'/12G)
C
C       OBTAIN COEFFICENTS FOR CURRENT SEGMENT BY MULTIPLYING FROM
C       LEFT BY SCRIPT-C INVERSE FOR CURRENT SEGMENT EVALUATED AT
C       RIGHT OF SEGMENT
C
      DO 134 I=1,NM
      I1=2*(I-1)
      C1=CCOS(QS(ISM,I)*X(IS))
      S1=CSIN(QS(ISM,I)*X(IS))
      SUM=C1*DUM(I1+1,1)-S1*DUM(I1+2,1)/QS(ISM,I)
      SUM2=S1*DUM(I1+1,1)+C1*DUM(I1+2,1)/QS(ISM,I)
      AM(I1+1,1)=SUM
      AM(I1+2,1)=SUM2
      ALF(IS,I)=SUM
  134 BET(IS,I)=SUM2
C        TYPE 251,(AM(I,1),I=1,NM2)
  251 FORMAT(' 251 AM'/12G)
C        TYPE 252,(ALF(IS,I),I=1,NM)
  252 FORMAT(' 252 ALF'/6G)
C        TYPE 253,(BET(IS,I),I=1,NM)
  253 FORMAT(' 253 BET'/6G)
  128 CONTINUE
C
C       RECOMPUTE COEFFICIENTS IN SOURCE SEGMENT
C
C        TYPE 620,(QS(1,I),I=1,NM)
  620 FORMAT(' 620 QS'/6G)
      DO 600 I=1,NM
      I1=2*(I-1)
      C1=CCOS(QS(1,I)*X(1))
      S1=CSIN(QS(1,I)*X(1))
      DUM(I1+1,1)=C1*AM(I1+1,1)+S1*AM(I1+2,1)
  600 DUM(I1+2,1)=QS(1,I)*(-S1*AM(I1+1,1)+C1*AM(I1+2,1))
C        TYPE 601,(DUM(I,1),I=1,NM2)
  601 FORMAT(' 601 DUM'/12G)
      DO 602 I=1,NM
      I1=2*(I-1)
      DO 602 J=1,2
      SUM=0.
      DO 603 K=1,NM
```

```
  603 SUM=SUM+US(1,I,K)*DUM(2*(K-1)+J,1)
  602 AM(I1+J,1)=SUM
C      TYPE 604,(AM(I,1),I=1,NM2)
  604 FORMAT(' 604 AM'/12G)
      X2=SQRT(X(1))
      DO 605 I=1,NM
      I1=2*(I-1)
      RHO=SQRT(XKN(I,1))*X(1)
      CALL HANK(RHO,H1,H1P,H2,H2P)
      XKTEM=RHO/X(1)
      A1=X2*H1
      A2=X2*H2
      A3=XKTEM*X2*H1P+H1/(2.*X2)
      A4=XKTEM*X2*H2P+H2/(2.*X2)
      A5=A1*A4-A2*A3
      TEM=A1
      A1=A4/A5
      A4=TEM/A5
      A2=-A2/A5
      A3=-A3/A5
      DUM(I1+1,1)=(A1*AM(I1+1,1)+A2*AM(I1+2,1))/A0(I)
  605 DUM(I1+2,1)=(A3*AM(I1+1,1)+A4*AM(I1+2,1))/A0(I)
C      TYPE 606,(DUM(I,1),I=1,NM2)
  606 FORMAT(' 606 DUM'/12G)
C
C     INPUT RANGES AND DEPTHS FOR FINDING PROP LOSS
C
   20 TYPE 21
   21 FORMAT('  INPUT 0 FOR RANGE VARIATION, 1 FOR DEPTH VARIATION')
      ACCEPT 2,IRZ
      IF(IRZ.EQ.1) GO TO 22
      TYPE 23
   23 FORMAT('  INITIAL RANGE, FINAL RANGE,',
     1' STEP SIZE')
      ACCEPT 2,R1,R2,DR
      R1=R1*1000.
      R2=R2*1000.
      DR=DR*1000.
      R1=R1-DR
      R=R1
      NR=(R2-R)/DR+.01
      NZ=1
      DZ=0.
      GO TO 24
   22 TYPE 25
   25 FORMAT('  INPUT RANGE, INITIAL DEPTH, FINAL DEPTH, STEP SIZE')
      ACCEPT 2,R,Z1,Z2,DZ
      Z1=Z1-DZ
      NZ=(Z2-Z1)/DZ+.01
      NR=1
      DR=0.
   24 DO 135 IR=1,NR
      R=R+DR
      ROUT=R/1000.
      DO 136 IS=1,NS
      IF(R.LT.X(IS)) GO TO 825
  136 CONTINUE
C
C     RANGE IS IN INFINITE SEGMENT
C
      IS=NSP
      ZB1=SQRT(ZB(1))
      ZBN=SQRT(ZB(NSP))
      DO 137 I=1,NM
      RHO=SQRT(XKN(I,NSP))*R
      CALL HANK(RHO,H1,H1P,H2,H2P)
```

```
  137 FR(I)=A(I)*H1/ZB1
      SUM=0.
      DO 750 I=1,NM
  750 SUM=SUM+FR(I)*UR(I,NSP)
      SUM=SUM/ZBN
      GO TO 26
  825 IF(IS.EQ.1) GO TO 27
C
C     RANGE IS NOT IN EITHER SOURCE OR INFINITE SEGMENT
C
      ISM=IS-1
      X2=SQRT(R)
      ZB1=SQRT(ZB(1))
      DO 138 I=1,NM
      SUM=0.
      DO 139 J=1,NM
      QR=QS(ISM,J)*R
  139 SUM=SUM+US(ISM,I,J)*(ALF(IS,J)*CCOS(QR)+BET(IS,J)*CSIN(QR))/X2
  138 FR(I)=SUM/ZB1
      GO TO 26
C
C     RANGE IS IN SOURCE SEGMENT
C
   27 ZB1=SQRT(ZB(1))
      DO 140 I=1,NM
      RHO=SQRT(XKN(I,1))*R
      CALL HANK(RHO,H1,H1P,H2,H2P)
  140 FR(I)=A0(I)*(ALF(1,I)*H1+BET(1,I)*H2)/ZB1
C
C     OUTPUT MODE INFORMATION
C
   26 IF(NOUT.EQ.0) GO TO 45
      J=1
      DO 150 I=1,NM
      IF(I.NE.IOUT(J)) GO TO 150
      I1=2*(J-1)
      G(I1+1)=CABS(FR(I))
      G(I1+2)=CLOG(FR(I)/G(I1+1))/EYE*180./PI
      IF(J.EQ.NOUT) GO TO 46
      J=J+1
  150 CONTINUE
   46 WRITE (23,47) ROUT,(G(I),I=1,NOUT2)
   47 FORMAT(F10.3,1P6E10.3/(10X,1P6E10.3))
C
C     COMPUTE PROP LOSS
C
   45 IF(IS.EQ.NSP) GO TO 751
      ISP=IS+1
      FRAC=(R-XNAG(IS))/(XNAG(ISP)-XNAG(IS))
      SUM=0.
      ZB1=SQRT(ZB(IS))
      ZB2=SQRT(ZB(ISP))
      DO 142 I=1,NM
  142 SUM=SUM+FR(I)*(UR(I,IS)/ZB1+FRAC*(UR(I,ISP)/ZB2-UR(I,IS)/ZB1))
  751 TL=-20.*ALOG10(4.*PI*CABS(SUM))
      IF(IRZ.EQ.1) GO TO 28
      IF(ROUT.NE.1..OR.ROUT.NE.10..OR.ROUT.NE.25..OR.ROUT.NE.50..OR.ROUT
     1.NE.75..OR.ROUT.NE.100.) GO TO 1234
      TYPE 29,ROUT,TL
   29 FORMAT(F13.5,F10.2)
 1234 CONTINUE
      WRITE (22,999) ROUT,TL
  999 FORMAT(F13.5,F15.6)
      GO TO 125
   28 TYPE 29,ZD,TL
      WRITE (22,29) ZD,TL
```

```
135 CONTINUE
    T=-1.
    WRITE (22,29) T
    TYPE 30
 30 FORMAT('  INPUT 0 FOR NEW RANGE OR DEPTH, 1 FOR END')
    ACCEPT 2,IEND
    IF(IEND.EQ.0) GO TO 20
    STOP
    END
```

```
TY MACL
    SUBROUTINE MACL(NM,NLP,NSP,NMODE,Z,ZS, [illegible] ,XKNS,UOS,URS,ZS,XKN,U[illegible]
```

```
TY NAGL
      SUBROUTINE NAGL(NM,NLP,NSP,NMODE,Z,ZB,XK,XKNS,UOS,URS,ZS,XKN,UO,
     1UR)
      DIMENSION Z(4),ZB(4),XK(4,4),XKN(15,4),UO(15),UR(15,4),NMODE(4)
      DIMENSION XKNS(15,4),UOS(15),URS(15,4)
      DOUBLE PRECISION UO,UR,XKN
      DO 100 ISP=1,NSP
      IM=NMODE(ISP)
      IF(IM.EQ.0) GO TO 300
      IF(ISP.EQ.1)READ(25,1)(UO(I),I=1,IM)
    1 FORMAT(100D)
C      IF(ISP.EQ.1)TYPE 237,(UO(I),I=1,IM)
  237 FORMAT(' 237 UO'/(3G))
      READ(27,1)(UR(I,ISP),I=1,IM)
C      TYPE 238,(UR(I,ISP),I=1,IM)
  238 FORMAT(' 238 UR'/3G)
      READ(28,1)(XKN(I,ISP),I=1,IM)
      DO 101 I=1,IM
  101 XKN(I,ISP)=XKN(I,ISP)**2
C      TYPE 239,(XKN(I,ISP),I=1,IM)
  239 FORMAT(' 239 XKN'/3G)
      IF(ZS.GT.0..OR.ISP.NE.1) GO TO 100
      DO 302 I=1,IM
  302 UO(I)=UR(I,1)
      GO TO 100
  300 IM=NMODE(1)
      NMODE(ISP)=NMODE(1)
      DO 301 I=1,IM
      UR(I,ISP)=UR(I,1)
  301 XKN(I,ISP)=XKN(I,1)
  100 CONTINUE
      DO 110 ISP=1,NSP
      IM=NMODE(ISP)
      DO 110 I=1,IM
      UOS(I)=UO(I)
      URS(I,ISP)=UR(I,ISP)
  110 XKNS(I,ISP)=XKN(I,ISP)
      RETURN
      END
```

```
TY FR\R\^C

.TTY FORM

.TY FORM
      SUBROUTINE FORM(IS,U,UI,Q,ICOUP,NMODE)
      COMMON /DEPTH/ XKN(15,4),
     1U0(15)
      COMMON /FORMC/ X(3),Z(4),ZB(4),NM,NL,NLM,NLP,XK(4,4),DEN(4)
     1,XNAG(4)
      DIMENSION AC(3),ST(15),CT(15),C(15,15),V(15,15),DUM(15,15),
     1S(15,15),SI(15,15),EIG(15),U(15,15),UI(15,15),Q(15),DMAT(15,15),
     2EMAT(120),DVEC(15,15),DEIG(15),WK(45),WK2(1480),SIG(15,15),
     3XMXEL(15,15),NMODE(4)
      COMPLEX SUM,C,V,DUM,S,SI,EIG,U,UI,Q,DMAT,EMAT,DVEC,SIG
      DOUBLE PRECISION XNAG
C
C     COMPUTES U=S*SIGMA*S1 AND ITS INVERSE AND THE EIGENVALUES Q OF
C     FINAL TRANSFORMATION
      MATSIZ=15
      NP=NM*(NM+1)/2
      NQ=3*NM
      NWK2=2*NM*(NM+1)
      ISM=IS-1
      ISP=IS+1
C
C     LINEARIZE KN**2 IN RANGE
C
      DO 100 I=1,NM
      ST(I)=(XKN(I,ISP)-XKN(I,IS))/(X(IS)-X(ISM))
  100 CT(I)=XKN(I,IS)-ST(I)*X(ISM)
C      TYPE 202,(ST(I),I=1,NM)
  202 FORMAT(' 202 ST'/3G)
C      TYPE 203,(CT(I),I=1,NM)
  203 FORMAT(' 203 CT'/3G)
C
C     LINEARIZE K**2 IN DEPTH AND RANGE
C
      DO 101 IL=1,NLM
  101 AC(IL)=(XK(IL+1,ISP)-XK(IL,ISP)-(XK(IL+1,IS)-XK(IL,IS)))/
     1((X(IS)-X(ISM))*(Z(IL+1)-Z(IL)))
      AC(NL)=((XK(NLP,ISP)-XK(NL,ISP))/(ZB(ISP)-Z(NL))
     1-(XK(NLP,IS)-XK(NL,IS))/(ZB(IS)-Z(NL)))/(X(IS)-X(ISM))
C      TYPE 204,(AC(I),I=1,NL)
  204 FORMAT(' 204 AC'/3G)
C
C     FORM MATRIX C ON PAGE 37
C
      CALL MXEL(NMODE,IS,XNAG,XMXEL,ZB)
C      TYPE 205,((XMXEL(I,J),J=1,NM),I=1,NM)
  205 FORMAT(' 205 XMXEL'/(3G))
      DO 104 I=2,NM
      IM=I-1
      DO 102 J=1,IM
      C(I,J)=-XMXEL(I,J)/((ST(I)-ST(J))*(X(IS)+X(ISM))/2.+CT(I)-CT(J))
  102 C(J,I)=-C(I,J)
  104 C(I,I)=0.
      C(1,1)=0.
C      TYPE 206,((C(I,J),J=1,NM),I=1,NM)
  206 FORMAT(' 206 C'/(6G))
C
C     FORM MATRIX V ON PAGE 36
C
      DO 105 I=1,NM
      DO 105 J=1,I
      SUM=0.
```

```
      DO 107 K=1,NM
      IF(K.EQ.I.OR.K.EQ.J) GO TO 107
      SUM=SUM+C(I,K)*C(K,J)
  107 CONTINUE
      SUM=-SUM
      IF(I.EQ.J) GO TO 2
      V(I,J)=-C(I,J)*(ST(I)-ST(J))/((ST(I)-ST(J))*(X(IS)+X(ISM))/2.
     1+CT(I)-CT(J))+SUM
      V(J,I)=-C(J,I)*(ST(J)-ST(I))/((ST(J)-ST(I))*(X(IS)+X(ISM))/2.
     1+CT(J)-CT(I))+SUM
      GO TO 105
    2 V(I,I)=SUM
  105 CONTINUE
      IF(ICOUP.EQ.1) GO TO 802
      DO 803 I=1,NM
      DO 804 J=1,NM
      U(I,J)=0.
      U(J,I)=0.
      UI(I,J)=0.
  804 UI(J,I)=0.
      U(I,I)=1.
      UI(I,I)=1.
      Q(I)=SQRT((XKN(I,IS)+XKN(I,ISP))/2.)
  803 CONTINUE
C      TYPE 805,((U(I,J),J=1,NM),I=1,NM)
  805 FORMAT(' 805 U'/(6G))
C      TYPE 806,((UI(I,J),J=1,NM),I=1,NM)
  806 FORMAT(' 806 UI'/(6G))
C      TYPE 807,(Q(I),I=1,NM)
  807 FORMAT(' 807 Q'/6G)
      RETURN
  802 CONTINUE
C      TYPE 207,((V(I,J),J=1,NM),I=1,NM)
  207 FORMAT(' 207 V'/(6G))
C
C     DIAGONALIZE C.
C     C IS ANTISYMMETRIC (ANTIHERMITIAN)...EIGENVALUES ARE IMAGINARY
C
      CALL HMDIAG(NM,NP,NQ,C,EIG,S,SI,DMAT,EMAT,DVEC,DEIG,WK)
C      TYPE 208,((S(I,J),J=1,NM),I=1,NM)
  208 FORMAT(' 208 S'/(6G))
C      TYPE 209,((SI(I,J),J=1,NM),I=1,NM)
  209 FORMAT(' 209 SI'/(6G))
C      TYPE 210,(EIG(I),I=1,NM)
  210 FORMAT(' 210 EIG'/6G)
C
C     DEFINE DIAGONAL MATRIX SIGMA
C
      DO 108 I=1,NM
      SIG(I,I)=CEXP(-EIG(I)*(X(IS)+X(ISM))/2.)
      IF(I.EQ.1) GO TO 108
      IM=I-1
      DO 300 J=1,IM
      SIG(I,J)=0.
  300 SIG(J,I)=0.
  108 CONTINUE
C      TYPE 211,((SIG(I,J),J=1,NM),I=1,NM)
  211 FORMAT(' 211 SIG'/(6G))
C
C     FORM MATRIX V+K
C
      DO 109 I=1,NM
  109 V(I,I)=V(I,I)+(XKN(I,IS)+XKN(I,ISP))/2.
C      TYPE 212,((V(I,J),J=1,NM),I=1,NM)
  212 FORMAT(' 212 V'/(6G))
C
```

```
C
C     FORM MATRIX (V+K)*S
C
      DO 110 I=1,NM
      DO 110 J=1,NM
      SUM=0.
      DO 111 K=1,NM
  111 SUM=SUM+V(I,K)*S(K,J)
  110 DUM(I,J)=SUM
C      TYPE 213,((DUM(I,J),J=1,NM),I=1,NM)
  213 FORMAT (' 213 DUM'/(6G))
C
C     FORM MATRIX T (CALL IT V TO SAVE SPACE) ON PAGE 38
C
      DO 114 I=1,NM
      DO 112 J=1,NM
      SUM=0.
      DO 113 K=1,NM
  113 SUM=SUM+SI(I,K)*DUM(K,J)
  112 V(I,J)=SUM*CEXP((EIG(I)-EIG(J))*(X(IS)+X(ISM))/2.)
  114 V(I,I)=V(I,I)-EIG(I)**2
C      TYPE 214,((V(I,J),J=1,NM),I=1,NM)
  214 FORMAT(' 214 V'/(6G))
C
C     DIAGONALIZE T(CALLED V IN CODE)
C
      CALL EIGCC(V,NM,MATSIZ,2,Q,U,MATSIZ,WK2,IER)
C      TYPE 888,(Q(I),I=1,NM)
  888 FORMAT(' 888 Q'/6G)
      DO 450 I=1,NM
  450 Q(I)=CSQRT(Q(I))
C      TYPE 215,(Q(I),I=1,NM)
  215 FORMAT(' 215 Q'/6G)
C      TYPE 216,((U(I,J),J=1,NM),I=1,NM)
  216 FORMAT(' 216 U'/(6G))
C
C     THE FOLLOWING CARD IS TEMPORARY TO CHECK PERFORMANCE OF EIGCC
C
      TYPE 400,WK2(1)
      IF(WK2(1).GT.1.) TYPE 400,WK2(1)
  400 FORMAT('  EIGCC PERFORMANCE=',1PE10.3)
C
C     FIND INVERSE OF U
C
      DO 410 I=1,NM
      DO 410 J=1,NM
  410 V(I,J)=U(I,J)
      CALL LEQT2C(V,NM,MATSIZ,DUM,1,MATSIZ,1,WK2,IER)
      IF(IER.NE.129) GO TO 412
      TYPE 411
  411 FORMAT('  U IS SINGULAR')
      STOP
  412 IF(IER.EQ.130) TYPE 413
  413 FORMAT('  ITERATION FAILED TO IMPROVE U INVERSE.  MATRIX IS',
     1' TOO ILL CONDITIONED')
      DO 414 IM=1,NM
      DO 415 JM=1,NM
  415 DUM(JM,1)=0.
      DUM(IM,1)=1.
      CALL LEQT2C(V,NM,MATSIZ,DUM,1,MATSIZ,2,WK2,IER)
      IF(IER.NE.129) GO TO 416
      TYPE 411
      STOP
  416 IF(IER.EQ.130) TYPE 413
      DO 417 JM=1,NM
  417 UI(JM,IM)=DUM(JM,1)
  414 CONTINUE
```

```
C       TYPE 217,((UI(I,J),J=1,NM),I=1,NM)
  217 FORMAT(' 217 UI'/(6G))
C
C      FORM MATRIX SIGMA*S1
       DO 115 I=1,NM
       DO 115 J=1,NM
  115 DUM(I,J)=SIG(I,I)*U(I,J)
C       TYPE 218,((DUM(I,J),J=1,NM),I=1,NM)
  218 FORMAT(' 218 DUM'/(6G))
C
C      FORM MATRIX U=S*SIGMA*S1
C
       DO 117 I=1,NM
       DO 117 J=1,NM
       SUM=0.
       DO 118 K=1,NM
  118 SUM=SUM+S(I,K)*DUM(K,J)
  117 U(I,J)=SUM
C       TYPE 219,((U(I,J),J=1,NM),I=1,NM)
  219 FORMAT(' 219 U'/(6G))
C
C      FORM MATRIX S1 INVERSE*SIGMA INVERSE
C
       DO 119 I=1,NM
       DO 119 J=1,NM
  119 DUM(I,J)=UI(I,J)/SIG(J,J)
C       TYPE 220,((DUM(I,J),J=1,NM),I=1,NM)
  220 FORMAT(' 220 DUM'/(6G))
C
C      FORM MATRIX UI INVERSE=S1 INVERSE*SIGMA INVERSE*S INVERSE
C
       DO 120 I=1,NM
       DO 120 J=1,NM
       SUM=0.
       DO 121 K=1,NM
  121 SUM=SUM+DUM(I,K)*SI(K,J)
  120 UI(I,J)=SUM
C       TYPE 221,((UI(I,J),J=1,NM),I=1,NM)
  221 FORMAT(' 221 UI'/(6G))
       RETURN
       END
```

```
TY COEF1
      SUBROUTINE COEF1(NM,AM,V,A,AO,ALF,BET,FRAC)
      DIMENSION AM(30,30),V(15),A(15),AO(15),ALF(3,15),BET(3,15),
     1C(15,15),V1(15),V2(15),V3(15)
      COMPLEX AM,V,A,AO,ALF,BET,C,SUM,SUM2
C
C     ROUTINE COMPUTES COEFFICIENTS IN INFINITE AND SOURCE SEGMENTS BY
C     MATCHING, USING AN ITERATIVE APPROACH.
C
      NC=10
      TYPE 911
  911 FORMAT(' INPUT MAXIMUM NUMBER OF ITERATIONS')
      ACCEPT 912,NC
  912 FORMAT(G)
C
C     FOR MODES WHICH DO NOT PROPAGATE AT SOURCE, V=0.
C     SET V=.00001*SMALLEST NONZERO V FOR ITERATION PROCESS.
C
      VMIN=CABS(V(1))
      DO 100 I=2,NM
      VT=CABS(V(I))
      IF(VT.LT.VMIN.AND.VT.GT.0.) VMIN=VT
  100 CONTINUE
      VMIN=VMIN*1.E-5
      DO 101 I=1,NM
      IF(V(I).EQ.(0.,0.)) V(I)=VMIN/(0.,1.)
      V1(I)=REAL(V(I))
      V2(I)=AIMAG(V(I))
      V3(I)=FRAC*CABS(V(I))
  101 CONTINUE
C       TYPE 229,(V(I),I=1,NM)
  229 FORMAT(' 229 V'/6G)
C       TYPE 230,(V1(I),I=1,NM)
  230 FORMAT(' 230 V1'/3G)
C       TYPE 231,(V2(I),I=1,NM)
  231 FORMAT(' 231 V2'/3G)
C       TYPE 232,(V3(I),I=1,NM)
  232 FORMAT(' 232 V3'/3G)
C
C     COMBINE ELEMENTS OF AM FOR USE IN EQUATION 4 OF PAGE NAGL.4
C
      DO 102 I=1,NM
      I1=2*(I-1)
      DO 102 J=1,NM
  102 C(I,J)=AM(I1+1,J)-AM(I1+2,J)
C       TYPE 233,((C(I,J),J=1,NM),I=1,NM)
  233 FORMAT(' 233 C'/(6G))
C
C     DEFINE A FOR ZEROTH ORDER (PAGE NAGL.5)
C
      DO 103 I=1,NM
  103 A(I)=V(I)/C(I,I)
C       TYPE 234,(A(I),I=1,NM)
  234 FORMAT(' 234 A'/6G)
C
C     IC=ITERATION COUNTER
C
      IC=1
      GO TO 50
C
C     DEFINE A FOR IC-TH ITERATION
C
    4 DO 104 I=1,NM
      SUM=0.
      DO 105 J=1,NM
      IF(J,EQ,I) GO TO 105
```

```
      SUM=SUM+C(I,J)*A(J)
  105 CONTINUE
  104 A(I)=(V(I)-SUM)/C(I,I)
C
C     THE FOLLOWING CARD IS TEMPORARY TO TEST ITERATION PROCEDURE
C
   50 IW=0
      DO 106 I=1,NM
      SUM=0.
      DO 107 J=1,NM
  107 SUM=SUM+C(I,J)*A(J)
      X1=REAL(SUM)
      X2=AIMAG(SUM)
C
C     THE FOLLOWING 2 CARDS SHOULD BE INCLUDED WHEN ITEERATION TEST
C     IS COMPLETED
C
C     IF(ABS(X1-V1(I)).GT.V3(I)) GO TO 1
C     IF(ABS(X2-V2(I)).GT.V3(I)) GO TO 1
C
C     THE FOLLOWING 3 CARDS ARE TEMPORARY TO TEST ITERATION PROCEDURE
C
      IF(ABS(X1-V1(I)).GT.V3(I)) IW=1
      IF(ABS(X2-V2(I)).GT.V3(I)) IW=1
C      TYPE 875,IC,I,A(I),X1,X2,V1(I),V2(I)
  875 FORMAT(2I3,1P2E15.7/6X,1P4E15.7)
  106 CONTINUE
C
C     THE FOLLOWING CARD SHOULD BE INCLUDED WHEN ITERATION TEST IS
C     COMPLETED
C
C     GO TO 2
C
C     THE FOLLOWING CARD IS TEMPORARY TO TEST ITERATION PROCEDURE
C
      IF(IW.EQ.0) GO TO 2
    1 IF(IC.EQ.NC) GO TO 3
C
C     START NEW ITERATION
C
      IC=IC+1
      GO TO 4
C
C     MAXIMUM NUMBER OF ITERATIONS REACHED WITHOUT CONVERGING
C
    3 TYPE 5
    5 FORMAT(//'  A NOT CONVERGED. SUM, V=')
      DO 109 I=1,NM
      SUM=0.
      DO 110 J=1,NM
  110 SUM=SUM+C(I,J)*A(J)
      TYPE 875,I,IC,A(I),SUM,V(I)
  109 CONTINUE
      TYPE 600
  600 FORMAT(' TYPE...-1 STOP'/
     1'          0 USE LAST VALUES FOR COEFFICIENTS'/
     2'          N ITERATE N MORE TIMES')
      ACCEPT 601,NC
  601 FORMAT(G)
      IF(NC) 602,2,603
  602 STOP
  603 IC=1
      GO TO 4
C
C     USING CONVERGED VALUES OF A (OR VALUES FROM NC-TH ITERATION),
C     DEFINE AO,AND ALFA,BETA FOR [illegible] INTERVAL
```

```
    2 DO 111 I=1,NM
      I1=2*(I-1)
      SUM=0.
      SUM2=0.
      DO 112 J=1,NM
      SUM=SUM+AM(I1+1,J)*A(J)
  112 SUM2=SUM2+AM(I1+2,J)*A(J)
      S=SQRT((CABS(SUM))**2+(CABS(SUM2))**2)
      A0(I)=S
      ALF(1,I)=SUM/S
  111 BET(1,I)=SUM2/S
C       TYPE 690,(A0(I),I=1,NM)
  690 FORMAT(' 690 A0'/6G)
C       TYPE 235,(ALF(1,I),I=1,NM)
  235 FORMAT(' 235 ALF'/6G)
C       TYPE 236,(BET(1,I),I=1,NM)
  236 FORMAT(' 236 BET'/6G)
      RETURN
      END
```

```
TY HMDIAG
      SUBROUTINE HMDIAG(N,NP,NQ,CMAT,CEIG,CVEC,CVECC,DMAT,EMAT,DVEC,
     1DEIG,WK)
      COMPLEX CMAT,CEIG,CVEC,CVECC,EYE
      COMPLEX DMAT,EMAT,DVEC
      DIMENSION CMAT(15,15),CVEC(15,15),CVECC(15,15),CEIG(15)
      DIMENSION DMAT(N,N),EMAT(NP),DVEC(N,N),DEIG(N),WK(NQ)
C
C     MUST DIVIDE BY EYE BECAUSE MATRIX IS ANTISYMMETRIC.
C     THIS MAKES MATRIX HERMITIAN WHICH CAN BE EASILY
C     DIAGONALIZED.  ORIGINAL MATRIX HAS EIGENVALUES WITH SAME MAGNITUDE
C     AS THE HERMITIAN MATRIX BUT ITS EIGENVALUES ARE PURELY IMAGINARY.
C     THE EIGENVECTORS ARE THE SAME.
C
      EYE=(0.,1.)
      DO 150 I=1,N
      DO 150 J=1,N
  150 DMAT(I,J)=CMAT(I,J)/EYE
      CALL VCVTCH(DMAT,N,N,EMAT)
      CALL EIGCH(EMAT,N,1,DEIG,DVEC,N,WK,IER)
C
C     MUST MULTIPLY BY EYE BECAUSE ANTISYMMETRIC MATRIX HAS IMAGINARY
C     EIGENVALUES
C
      DO 151 I=1,N
      CEIG(I)=DEIG(I)*EYE
      DO 151 J=1,N
      CVEC(I,J)=DVEC(I,J)
  151 CVECC(J,I)=CONJG(CVEC(I,J))
      RETURN
      END
```

```
TY MXEL
            SUBROUTINE MXEL(NMODE,NPRS,D,OXEL,OZB)
            IMPLICIT DOUBLE PRECISION (A-H,P-Z)
            DIMENSION X(15,4),A(15,4),B(15,4),U(15,4),UP(15,4),UINT(
15)
            DIMENSION D(4),SL(4),AL(4),S(15,15),OXEL(15,15),OZB(4),N
MODE(4)
            M=NMODE(NPRS+NC-1)
            DO 36 NC=1,2
            DO 35 NP=1,NPRS+NC-1
            DO 30 J=1,M
            READ(21,40) NB
            DO 30 I=1,NB
30          READ(21,10) X(J,I),A(J,I),B(J,I),U(J,I),UP(J,I)
10          FORMAT(5D)
40          FORMAT(I4)
C           TYPE 80,((U(J,I),I=1,4),J=1,M)
80          FORMAT(4D16.7)
C           TYPE 80,((UP(J,I),I=1,4),J=1,M)
35          CONTINUE
            IF(NC.EQ.2) GO TO 39
            READ(21,40) NB1
            DO 31 I=1,NB1
31          READ(21,10) C1,AL(I),C2,C3,C4
            DO 38 KL=2,NB
38          SL(KL)=(AL(KL)**3-A(1,KL)**3)/
         1  (D(NPRS+1)-D(NPRS))
39          CONTINUE
            CLOSE(UNIT=21)
            DO 100 J=2,M
            DO 100 K=1,J-1
            IF(NC.EQ.1) S(J,K)=0.
            DO 110 I=2,NB
            BD=(B(J,I)-B(K,I))*A(J,I)**2
            BS=B(J,I)+B(K,I)
            V=A(J,I)*(BS+2.*X(J,I))*U(J,I)*U(K,I)+2.*UP(J,I)*UP(K,I)
            W=(BD*X(J,I)-2.*A(J,I)/BD)*(UP(J,I)*U(K,I)-U(J,I)*UP(K,I
))
            Y=A(J,I)*(BS+2.*X(J,I-1))*U(J,I-1)*U(K,I-1)+
         1  2.*UP(J,I-1)*UP(K,I-1)*(A(J,I-1)/A(J,I))**2
            Z=(X(J,I-1)*BD-2.*A(J,I)/BD)*(UP(J,I-1)*U(K,I-1)-
         1  U(J,I-1)*UP(K,I-1))*A(J,I-1)/A(J,I)
            S(J,K)=S(J,K)+(V+W-Y-Z)/BD/BD*SL(I)/2./(OZB(NPRS+NC-1))*
*2
            OXEL(J,K)=S(J,K)
105         FORMAT(2I4,D)
110         CONTINUE
100         CONTINUE
36          CONTINUE
            CONTINUE
            RETURN
            END
```

```
TY HANK
      SUBROUTINE HANK(RHO,H1,H1P,H2,H2P)
      COMPLEX H1,H1P,H2,H2P
      CALL HANLG(RHO,H1,H1P,H2,H2P)
      RETURN
      END
```

```
TY HANLG
            SUBROUTINE HANLG(X,H1,H1P,H2,H2P)
            REAL J0,J1
            DOUBLE PRECISION PI,Z,A,P,Q,R,S,X
            COMPLEX H1,H1P,H2,H2P
            PI=3.14159265359
            Z=8*X
            Z1=DSQRT(2/PI/X)
            A=X-PI/4
            P=1-4.5/Z**2+9*25*49/24./Z**4
            Q=-1/Z+9*25/6./Z**3
            R=1+7.5/Z**2
            S=3/Z-9*35/6./Z**3
            J0=Z1*SNGL(P*DCOS(A)-Q*DSIN(A))
            Y0=Z1*SNGL(P*DSIN(A)+Q*DCOS(A))
            H1=J0+(0.,1.)*Y0
      50    FORMAT(2G)
            J0=-Z1*(R*DSIN(A)+S*DCOS(A))
            Y0=Z1*(R*DCOS(A)-S*DSIN(A))
            H1P=J0+(0.,1.)*Y0
            H2=CONJG(H1)
            H2P=CONJG(H1P)
            RETURN
            END
```

TY DAIRY

```
      SUBROUTINE DAIRY(DX,AI,AIP,BI,BIP)
        IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C FOR DOUBLE PRECISION ARGUMENTS, THIS ROUTINE CALCULATES THE AIRY
C     FUNCTION AI(X) AND ITS DERIVATIVE AIP(X).  IT ALSO FINDS
C     THE OTHER REAL LINEARLY INDEPENDENT SOLUTION BI(X) AND
C     ITS DERIVATIVE BIP(X).
C     THE DEFINITIONS AND NORMALIZATIONS ARE AS IN NBS HANDBOOK
C     OF MATHEMATICAL FUNCTIONS,P.446
C     THE METHODS USED ARE POWER SERIES EXPANSION FOR SMALL X
C     AND GAUSSIAN INTEGRATION FOR LARGE X
      DIMENSION X(16),W(16),XSQ(16)
C     DOUBLE PRECISION DX,AI,AIP,BI,BIP
C     DOUBLE PRECISION    XS ,XCUBE,AISUM,AIPSUM
C     DOUBLE PRECISION DF,DFP,DG,DGP
C     DOUBLE PRECISION FJM2,FJM1,FJ,FJP1,FJP2,FACTOR
C     DOUBLE PRECISION C1,C2,ROOT3
C     DOUBLE PRECISION DZETA,DARG,DROOTX
C     DOUBLE PRECISION ROOT4X,S,CO,RATIO,EFAC,ZETASQ
C     DOUBLE PRECISION SUMR,SUMI,SUMRP,SUMIP,TERMR,TERMI
C     DOUBLE PRECISION DZERO,DA,DB,DEN,ONE
C     DOUBLE PRECISION X,W,XSQ
C     DOUBLE PRECISION RSQ,TEMP,    RTPI,RTPI2
C     DOUBLE PRECISION TERMA,TERMB
      LOGICAL NEEDBI
      DATA DZERO,ONE /0.0D0,1.0D0/
      DATA ROOT3/1.732050807568877D0/
      DATA C1,C2 /.355028053887817D0, .258819403792807D0/
      DATA RTPI /.282094791773878D0/
      DATA RTPI2/.564189583547756D0/
C POSITIONS AND WEIGHTS FOR 10-TERM SUM FOR AIRY FUNCTIONS
        DATA W( 1) /  3.154[illegible]762964787D-14/
        DATA W( 2) /  6.6394[illegible]19584921D-11/
        DATA W( 3) /  1.7583[illegible]17435669D-08/
```

```
      DATA W( 4) /  1.3717592370435815D-06/
      DATA W( 5) /  4.4350966637284350D-05/
      DATA W( 6) /  7.1555010917718255D-04/
      DATA W( 7) /  6.4889566103335381D-03/
      DATA W( 8) /  3.6440415875773282D-02/
      DATA W( 9) /  1.4399792418590999D-01/
      DATA W(10) /  8.1231141336261486D-01/
      DATA X( 1) /  1.4083081072180964D+01/
      DATA X( 2) /  1.0214885479197331D+01/
      DATA X( 3) /  7.4416018450450930D+00/
      DATA X( 4) /  5.3070943061781927D+00/
      DATA X( 5) /  3.6340135029132462D+00/
      DATA X( 6) /  2.3310652303052450D+00/
      DATA X( 7) /  1.3447970824609268D+00/
      DATA X( 8) /  6.4188858369567296D-01/
      DATA X( 9) /  2.0100345998121046D-01/
      DATA X(10) /  8.0594359172052833D-03/
      DATA XSQ( 1) /0.198333172485621700D 03/
      DATA XSQ( 2) /0.104343885353116500D 03/
      DATA XSQ( 3) /0.553774380201781700D 02/
      DATA XSQ( 4) /0.281652499746689900D 02/
      DATA XSQ( 5) /0.132060541393558000D 02/
      DATA XSQ( 6) /0.543386510793804400D 01/
      DATA XSQ( 7) /0.180847919299542000D 01/
      DATA XSQ( 8) /0.412020953878836900D 00/
      DATA XSQ( 9) /0.404023909244180700D-01/
      DATA XSQ(10) /0.649545073035383900D-04/
C     POSITIONS AND WEIGHTS FOR  4-TERM SUM FOR AIRY FUNCTIONS
      DATA W(11) /  4.7763903057577263D-05/
      DATA W(12) /  4.9914306432910959D-03/
      DATA W(13) /  8.6167846973840312D-02/
      DATA W(14) /  9.0879095845981102D-01/
      DATA X(11) /  3.9198329554455091D+00/
```

```
      DATA X(12) /  1.6915619004823504D+00/
      DATA X(13) /  5.0275532467263018D-01/
      DATA X(14) /  1.9247060562015692D-02/
     DATA XSQ(11) /0.15365090398596670D 02/
     DATA XSQ(12) /0.28613816631634610D 01/
     DATA XSQ(13) /0.25276291648668180D 00/
     DATA XSQ(14) /0.37044934027789980D-03/
C    POSITIONS AND WEIGHTS FOR  2-TERM SUM FOR AIRY FUNCTIONS
      DATA W(15) /  9.6807280595773604D-01/
      DATA W(16) /  3.1927194042263958D-02/
      DATA X(15) /  3.6800601866153044D-02/
      DATA X(16) /  1.0592469382112375D+00/
     DATA XSQ(15) /0.13542842977111070D-02/
     DATA XSQ(16) /0.11220040761098810D 01/
     IF(DX.LT.-5.0D0) GO TO 100
     NEEDBI=.FALSE.
     IF(DX.GT.3.7D0) GO TO 200
C    THIS ROUTE FOR SMALLX, USING POWER SERIES.
C    INITIALIZE
  10 XS   = DX*DX
     XCUBE = XS *DX
     XS   = XS *0.5D0
     DF = C1
     DFP = C1*XS
     DG = C2*DX
     DGP = C2
     AISUM = DF - DG
     AIPSUM = DFP - DGP
     BI = DF + DG
     BIP = DFP + DGP
     FJM2=-2.0D0
  20 FJM2=FJM2+3.0D0
     FJM1=FJM2+ONE
```

```
      FJ=FJM1+ONE
      FJP1=FJ+ONE
      FJP2=FJP1+ONE
      RATIO = XCUBE/FJ
      DF = DF*RATIO/FJM1
      DFP = DFP*RATIO/FJP2
      DG = DG*RATIO/FJP1
      DGP = DGP*RATIO/FJM2
      BI = BI + (DF+DG)
      BIP = BIP + (DFP+DGP)
      IF(NEEDBI) GO TO 80
      AISUM = AISUM + (DF-DG)
      AIPSUM = AIPSUM + (DFP-DGP)
C     CONVERGENCE TEST
80    IF(DABS(DF).GT.1.0D-16) GO TO 20
C     CONVERGENCE. COMPUTE FUNCTIONS
99    BI = ROOT3*BI
      BIP = ROOT3*BIP
C THIS RETURNS IF X IS BETWEEN 3.7 AND 8.0, SINCE IN SUCH CASES MORE
C ACCURATE VALUES OF AI AND AIP HAVE ALREADY BEEN FOUND BY GAUSSIAN
C INTEGRATION
      IF(NEEDBI)RETURN
      AI = AISUM
      AIP = AIPSUM
      RETURN
C GAUSSIAN INTEGRATION FOR LARGE NEGATIVE X
100   DROOTX = DSQRT(-DX)
      ROOT4X = DSQRT(DROOTX)
      DZETA = -.6666666666666667*DX*DROOTX
      DARG = DZETA - .7853981633974483
      SUMR = DZERO
      SUMI = DZERO
      SUMRP = DZERO
```

```
      SUMIP = DZERO
C   TEST TO SEE HOW MANY TERMS ARE NEEDED IN GAUSSIAN INTEGRATION
      IF(DX.LT.(-200.D0)) GO TO 140
      IF(DX.LT.(-15.D0)) GO TO 130
C   THIS CASE FOR DX BETWEEN -5.0 AND -15.0
      LIMLO=1
      LIMHI=10
      GO TO 149
C   THIS CASE FOR DX BETWEEN -15.0 AND -200.
130   LIMLO=11
      LIMHI=14
      GO TO 149
C   THIS CASE FOR DX.LT.-200.
140   LIMLO=15
      LIMHI=16
149   ZETASQ=DZETA**2
      DO 150 K=LIMLO,LIMHI
      TERMR=W(K)/((ZETASQ+XSQ(K))**2)
      SUMR = SUMR + TERMR
      TERMR=TERMR*X(K)
      SUMI=SUMI+TERMR
      TERMR=TERMR*X(K)
      SUMRP=SUMRP+TERMR
150   SUMIP=SUMIP+TERMR*X(K)
      SUMR=(SUMR*ZETASQ+SUMRP)*ZETASQ
      TEMP=SUMI*ZETASQ
      SUMI=(TEMP+SUMIP)*DZETA
      SUMRP=SUMRP*DZETA
      SUMIP=SUMIP-TEMP
C   FORM AIRY FUNCTIONS
196   S = DSIN(DARG)
      CO = DCOS(DARG)
      RATIO = RTPI2/ROOT4X
```

```
      AI = RATIO*(CO*SUMR + S*SUMI)
      BI = RATIO*(CO*SUMI - S*SUMR)
      SUMRP=SUMRP+SUMRP
      RATIO = -.25D0/DX
      FACTOR = -RTPI2*ROOT4X
      AIP = RATIO*AI - DROOTX*BI + FACTOR*(CO*SUMRP+S*SUMIP)
      BIP = RATIO*BI + DROOTX*AI + FACTOR*(CO*SUMIP-S*SUMRP)
      RETURN
C   GAUSSIAN INTEGRATION FOR LARGE POSITIVE X
200   DROOTX = DSQRT(DX)
      DZETA = .6666666666666667*DX*DROOTX
      EFAC = DEXP(-DZETA)
      ROOT4X = DSQRT(DROOTX)
      AI = DZERO
      BI = DZERO
      AIP = DZERO
      BIP = DZERO
      IF(DX.LT.8.0D0) NEEDBI=.TRUE.
C  TEST TO SEE HOW MANY TERMS ARE NEEDED IN GAUSSIAN INTEGRATION
      IF(DX.GT.15.0D0) GO TO 230
C  THIS CASE FOR DX BETWEEN 3.7 AND 15.
      LIMLO=1
      LIMHI=10
      GO TO 249
C  THIS CASE FOR DX GREATER THAN 15.
230   LIMLO=11
      LIMHI=14
249   DO 250 K=LIMLO,LIMHI
      DA=DZETA+X(K)
      TERMA = W(K)/DA
      AI = AI + TERMA
      AIP=AIP+TERMA*X(K)/DA
      IF(NEEDBI) GO TO 250
```

```
      DB=DZETA-X(K)
      TERMB = W(K)/DB
    , BI = BI + TERMB
      BIP=BIP+TERMB*X(K)/DB
250   CONTINUE
C  FORM FUNCTIONS
      FACTOR=RTPI*DZETA/ROOT4X
      RATIO = 0.25D0/DX
      AI=AI*EFAC*FACTOR
      AIP=-(DROOTX+RATIO)*AI+RTPI*ROOT4X*EFAC*AIP
C  THIS IS SATISFIED ONLY FOR X BETWEEN 3.7 AND 8.0  IN THESE CASES
C  THE BI AND BIP ABOUT TO BE COMPUTED ARE NOT SUFFICIENTLY ACCURATE.
C  THUS RETURN TO POWER SERIES FOR BI AND BIP.
      IF(NEEDBI) GO TO 10
      FACTOR=FACTOR+FACTOR
      BI=BI*FACTOR/EFAC
      BIP=(DROOTX-RATIO)*BI-RTPI2*ROOT4X*BIP/EFAC
      RETURN
      END
```

```
TY FLT
      DIMENSION Y(150),IX(0/1000),IY(0/1000)
      TYPE 100
  100 FORMAT('  PLOT TYPE=-1...NO MORE PLOTS'/
     114X,'0...TRANSMISSION LOSS'/
     212X,'N,I...N=MODE NUMBER, I=0(MAGNITUDE) OR 1(PHASE)')
    1 TYPE 101
  101 FORMAT('  ENTER NUMBER OF MODES ON TAPE, PLOT TYPE')
      ACCEPT 102,NM,IPL,IMP
  102 FORMAT(10G)
      NM=2*NM
      IF(IPL.EQ.-1) STOP
      TYPE 300
  300 FORMAT(' TYPE TAPE NUMBER')
      ACCEPT 301,NT
  301 FORMAT(G)
      IN=2*(IPL-1)+IMP+1
      TYPE 103
  103 FORMAT('  ENTER 0 FOR LINE, 1 FOR POINTS')
      ACCEPT 102,ILPT
      TYPE 104
  104 FORMAT('  INPUT NTOT,NMIN,NMAX,XMIN,XMAX,YMIN,YMAX,NX,NY')
      ACCEPT 102,NTOT,NMIN,NMAX,XMIN,XMAX,YMIN,YMAX,NX,NY
      IF(IPL.NE.0) GO TO 500
      YTEM=YMIN
      YMIN=-YMAX
      YMAX=-YTEM
  500 CONTINUE
      DX=9999./NX
      DY=9999./NY
      TYPE 130
  130 FORMAT(' PLTP')
      DO 10 I=0,NX
      IY(I)=0
      IX(I)=DX*I
   10 TYPE 30,IX(I),IY(I)
   30 FORMAT(2I10)
      DO 11 I=0,NY
      IX(I)=0
      IY(I)=DY*I
   11 TYPE 30,IX(I),IY(I)
      TYPE 70
   70 FORMAT(' PLTT')
      REWIND NT
      DX=9999./(XMAX-XMIN)
      DY=9999./(YMAX-YMIN)
      IF(IPL.NE.0) GO TO 200
      DO 12 I=1,NMIN
      READ (NT,105) X,Y(1)
  105 FORMAT(F13.5,F15.6)
      IX(I)=DX*(X-XMIN)
      Y(1)=-Y(1)
   12 IY(1)=DY*(Y(1)-YMIN)
      GO TO 201
  200 DO 202 I=1,NMIN
      READ (NT,203) X,(Y(J),J=1,NM)
  203 FORMAT(F10.3,1P6E10.3/(10X,1P6E10.3))
      IX(I)=DX*(X-XMIN)
  202 IY(I)=DY*(Y(IN)-YMIN)
  201 TYPE 130
      TYPE 30,IX(NMIN),IY(NMIN)
      TYPE 70
      IF(IPL.NE.0) GO TO 204
      DO 13 I=NMIN+1,NMAX
      READ (NT,105) X,Y(1)
      IX(I)=DX*(X-XMIN)
```

```
      Y(1)=-Y(1)
   13 IY(I)=DY*(Y(1)-YMIN)
      GO TO 205
  204 DO 206 I=NMIN+1,NMAX
      READ (NT,203) X,(Y(J),J=1,NM)
      IX(I)=DX*(X-XMIN)
  206 IY(I)=DY*(Y(IN)-YMIN)
  205 IF(ILPT.EQ.1) GO TO 222
      TYPE 60
   60 FORMAT(' PLTL')
      GO TO 223
  222 TYPE 130
  223 CONTINUE
      DO 14 I=NMIN,NMAX
   14 TYPE 30,IX(I),IY(I)
      TYPE 70
      GO TO 1
      END
```

TO RUN DEPTH PROGRAM.
?TO?

.EX ZAR25,DAIRY,SELAUT,PARAM,SERV,SERD,SINCOS

input: tape 38
output: 21,25,27,28
also uses 24,26,29

^C

.TO RUN RANGE PROGRAM:
?TO?

.LOAD/F10 GPC,NAGL,FORM,COEF1,HMDIAG,MXEL,HANK,HANLG,DAIRY,STA:IMSL/LIB,
STA:IMSL/LIB,STA:IMSL/LIB,STA:IMSL/LIB

input 21,25,27,28
output 22,23

^C

.TO RUN PLOT PROGRAM:
?TO?

.LOAD/F10 PLT

input 22,23

^C